A Review of Pseudo-Labeling for Computer Vision

PATRICK KAGE*, School of Informatics, The University of Edinburgh, UK
JAY C. ROTHENBERGER*, School of Computer Science, The University of Oklahoma, USA
PAVLOS ANDREADIS, School of Informatics, The University of Edinburgh, UK
DIMITRIOS I. DIOCHNOS, School of Computer Science, The University of Oklahoma, USA

Deep neural models have achieved state-of-the-art performance on a wide range of problems in computer science, especially in computer vision. However, deep neural networks often require large datasets of labeled samples to generalize effectively. An important area of active research is *semi-supervised learning*, which attempts to instead utilize large quantities of (easily acquired) unlabeled samples. One family of methods in this space is *pseudo-labeling*, a class of algorithms that use model outputs to assign labels to unlabeled samples which are then used as labeled samples during training. Such assigned labels, called *pseudo-labels*, are most commonly associated with the field of semi-supervised learning. In this work, we explore a broader interpretation of pseudo-labels within both self-supervised and unsupervised methods. After a thorough treatment of pseudo-labeling in these areas, we draw the connection between them and identify commonalities between fields, as well as new directions where advancements in one area would likely benefit others, such as curriculum learning and self-supervised regularization.

JAIR Track: Surveys

JAIR Associate Editor: Ivor Tsang

JAIR Reference Format:
Patrick Kage*, Jay C. Rothenberger*, Pavlos Andreadis, and Dimitrios I. Diochnos. 2026. A Review of Pseudo-Labeling for Computer Vision. *Journal of Artificial Intelligence Research* 85, Article 26 (March 2026), 32 pages. DOI: 10.1613/jair.1.19656

1 Introduction

Deep neural networks have emerged as transformative tools especially in natural language processing and computer vision due to their impressive performance on a broad spectrum of tasks requiring generalization. However, a significant limitation of these systems is their requirement for a large set of labeled data for training. This becomes particularly challenging in niche domains such as scientific fields where human annotation requires domain experts, making the dataset curation process laborious and often prohibitively expensive. This review will focus on a specific methodology, *pseudo-labeling* (PL), within the broader field of semi-supervised and unsupervised computer vision tasks. In Figure 1 we provide a taxonomy of PL, and in Table 1 we give a performance comparison for various methods of PL.

* Equal contribution

Authors' Contact Information: Patrick Kage*, ORCID: 0000-0002-5639-1237, p.kage@ed.ac.uk, School of Informatics, The University of Edinburgh, Edinburgh, Midlothian, UK; Jay C. Rothenberger*, ORCID: 0009-0007-2530-4667, jay.c.rothenberger@ou.edu, School of Computer Science, The University of Oklahoma, Norman, Oklahoma, USA; Pavlos Andreadis, ORCID: 0000-0001-9160-4851, pavlos.andreadis@ed.ac.uk, School of Informatics, The University of Edinburgh, Edinburgh, Midlothian, UK; Dimitrios I. Diochnos, ORCID: 0000-0002-2934-606X, diochnos@ou.edu, School of Computer Science, The University of Oklahoma, Norman, Oklahoma, USA.

DOI: 10.1613/jair.1.19656

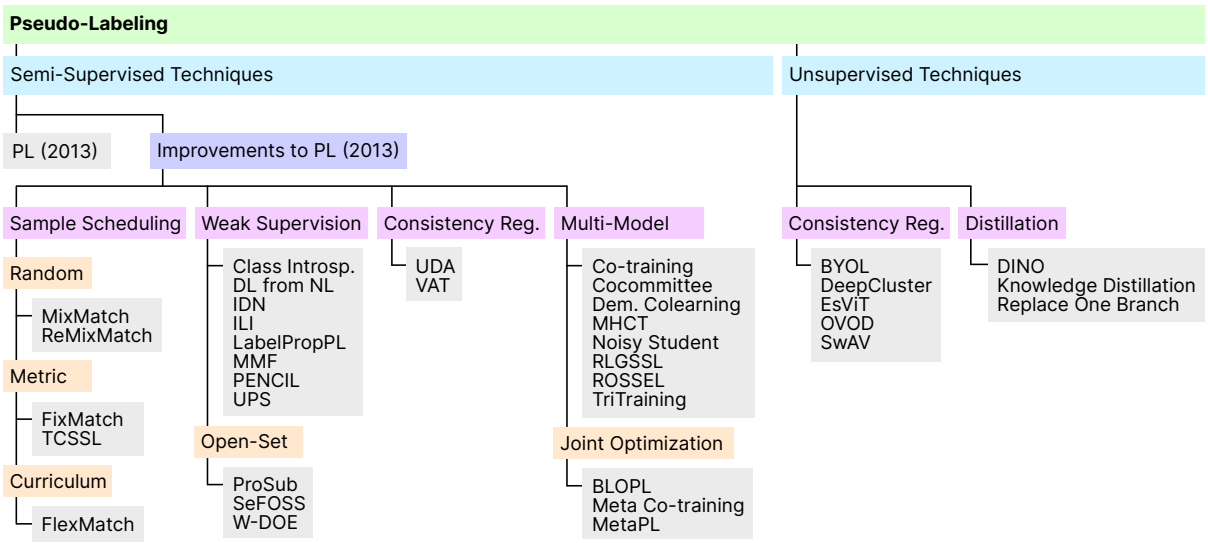

Fig. 1. Family tree of pseudo-labeling methods. Please see Appendix A for a table with section links and references. The methods shown in the figure from left to right are presented in the text in that order. Section 3 is devoted to semi-supervised learning methods and Section 4 to unsupervised learning methods. Furthermore, the third level split in the figure indicates in which subsection we discuss the various methods. For example, consistency regularization methods in SSL are discussed in Section 3.3, while consistency regularization methods in UL are discussed in Section 4.1. In Table 1 we give a performance comparison of relevant methods.

Semi-supervised learning (SSL) aims to address the missing-labeling problem by introducing strategies to incorporate unlabeled samples into the training process, allowing for training with only a small percentage of labeled samples (van Engelen and Hoos, 2020; Chapelle et al., 2006; Zhu and Goldberg, 2009; Prakash and Nithya, 2014; Pise and Kulkarni, 2008; Ouali et al., 2020; Yang et al., 2023; Amini et al., 2025). In recent years so-called "scaling laws" indicate that more data and larger models lead to better performance, thus driven by a demand for data to train ever-larger models SSL has become an important focus of research within the machine learning research community. There are several extant approaches to semi-supervised learning, including consistency-regularization-based approaches, fully generative approaches, and entropy minimization Ouali et al. (2020)—however, this review will focus on pseudo-labeling within this field.

Pseudo-labels (also known as *proxy labels*) have primarily been considered in the context of SSL. However, pseudo-labeling can also be seen across sub-fields of *unsupervised learning* (UL), particularly *self-supervised* learning (Caron et al., 2020) and *knowledge distillation* (Hinton et al., 2015). We formalize a definition of pseudo-labels which unifies these different areas.

In this analysis, we investigate the ways in which PL techniques across SSL and UL compare and contrast with each other. We show that PL techniques largely share one or more of the following characteristics:

- *self-sharpening*, where a single model is used to create the pseudo-label from a single view of a set of data points,
- *multi-view learning*, where single samples are represented in multiple ways to promote learning generalizable patterns, and
- *multi-model learning*, where multiple models are used during the learning process (e.g., in a student-teacher setting).

Table 1. Performance of traditional pseudo-labeling for semi-supervised learning approaches and related methods in unsupervised/self-supervised learning. Unsupervised and self-supervised methods are frequently leveraged to learn representations from unlabeled data and then those representations enable learning from few examples. This combination of methods is semi-supervised, and that semi-supervised performance is what is shown in the second part of the table. Models trained may differ between algorithms. All CIFAR-10-4k models were trained using a WideResNet-28-2. Nearly all ImageNet-10% models were trained with ViT-L. Approaches which use a ResNet-50 backbone are marked with (*) and approaches using a Swin-B backbone are marked with (†).

Method	Reference	CIFAR-10-4k	ImageNet-10%
Semi-Supervised			
BLOPL	Heidari and Guo (2025)	96.88	
RLGSSL	Heidari et al. (2024)	96.48	
MHCT	Chen et al. (2022)	96.16	
MPL	Pham et al. (2021)	96.11	73.89
FlexMatch	Zhang et al. (2021)	95.81	
FixMatch	Sohn et al. (2020)	95.74	71.5
ReMixMatch	Berthelot et al. (2019a)	94.86	
Meta-Semi	Wang et al. (2020)	93.90	
UPS	Rizve et al. (2021)	93.61	
MixMatch	Berthelot et al. (2019b)	92.71	
ICT	Verma et al. (2022)	92.71	
Deep Co-Training	Qiao et al. (2018)	91.65	53.5
Π-Model	Laine and Aila (2017)	87.84	
Mean Teacher	Tarvainen and Valpola (2017)	84.13	
Meta Co-Training	Rothenberger and Diochnos (2023)		85.8
Unsupervised/Self-Supervised			
TCSSL	Zhou et al. (2020)	94.97	
UDA	Xie et al. (2020a)	94.53	68.07
VAT	Miyato et al. (2019)	86.87	
CLIP	Radford et al. (2021)		84.7
DINOv2	Caron et al. (2021)		82.9
EsViT	Li et al. (2022)		†74.4
SwAV	Caron et al. (2020)		70.2
SimCLR	Chen et al. (2020)		*69.3
BYOL	Grill et al. (2020)		*68.8

We outline directions for future work that lie at the intersection of these areas, and directions illuminated in one area by considering established methods in the other.

Pseudo-labels within semi-supervised learning. Pseudo-labeling was first applied to deep learning for computer vision in Lee (2013).[1] Most techniques within SSL improve on this paper; see Figure 1. In Lee (2013), it is argued that PL is equivalent to *entropy minimization* (EM), maximizing the margin of the decision boundary in a

[1]There are self-training methods that predate the work of Lee (2013); e.g. Yarowsky (1995); Nigam et al. (2000). Here we follow the common usage of pseudo-labeling in deep learning literature.

classification problem by ensuring examples with similar labels lie close to each other on the underlying data manifold, and far from other examples. Though not explicitly mentioned in Lee (2013), this is an effect of two of the three assumptions inherent to SSL regimes: the *low density separation assumption* that samples are embedded in high-density regions separated by low-density regions and the *cluster assumption* that points within high density regions share the same class label (Chapelle et al., 2006).[2]

Subsequently, the literature (Cascante-Bonilla et al., 2020; Sohn et al., 2020; Berthelot et al., 2019b,a; Zhou et al., 2020; Xie et al., 2020a; Miyato et al., 2019; Shi et al., 2018; Iscen et al., 2019) includes the notion of *consistency regularization* (CR), in which examples that have been slightly perturbed (e.g., by an augmentation or adversarial additive perturbation) should lie close together in any embedding space or output distribution space produced by the model. CR and EM form strong theoretical arguments for PL with techniques using the average of predicted probabilities for augmentations to assign pseudo-labels to achieve *sample scheduling* or *curriculum learning* (CL) and most approaches choosing to assign "hard" (one-hot) pseudo-labels to examples to encourage EM (Berthelot et al., 2019b; Sohn et al., 2020; Berthelot et al., 2019a; Zhou et al., 2020; Zhang et al., 2021). When it can be assumed that examples form clusters with high internal consistency and low density separation, then it may be appropriate to perform *label propagation* with methods such as Iscen et al. (2019); Zhu and Ghahramani (2002); Shi et al. (2018). It is natural to think of the pseudo-label assigned by some model as *weak supervision*, and there are a variety of methods that leverage techniques for utilizing unreliable labels (Haase-Schütz et al., 2020; Tong Xiao et al., 2015; Yi et al., 2022; Wang et al., 2022; Wallin et al., 2023, 2024). Furthermore, there are methods that utilize multiple models in the hopes that each model will fail independently when predicting pseudo-labels for a given instance and that this independence of failure can be leveraged to produce high-quality pseudo-labels (Pham et al., 2021; Xie et al., 2020b; Blum and Mitchell, 1998; Zhou and Goldman, 2004; Zhou and Li, 2005; Yan et al., 2016; Chen et al., 2022; Rothenberger and Diochnos, 2023; Heidari and Guo, 2025; Heidari et al., 2024). Finally, we note that some of the above-mentioned lines of work combine multiple ideas. However, in Figure 1 we place these lines of work in the branch that we identify as most fitting for their contribution, given the taxonomy that we provide.

Pseudo-labels within unsupervised learning. Pseudo-labels have been applied in sub-fields of unsupervised learning. Self-supervised learning (Caron et al., 2021; Li et al., 2022; Caron et al., 2020; Chen et al., 2020; Radford et al., 2021; Zhai et al., 2023; Jia et al., 2021; Balestriero et al., 2023) is typically some form of CR optionally combined with contrastive learning. Knowledge distillation (Hinton et al., 2015; Duval et al., 2023; Zhang et al., 2019; Lopes et al., 2017; Wen et al., 2021) also makes use of pseudo labels to distill the knowledge of one model into another. These subfields are not traditionally considered pseudo-labeling, but they all make use of one or more neural networks which provide some notion of supervision. In some cases, this supervision corresponds to an approximation of a ground truth classification function (Hinton et al., 2015; Miyato et al., 2019; Xie et al., 2020a; Zhang et al., 2019; Wen et al., 2021), in some cases this supervision serves only to transfer knowledge from one network (sometimes referred to as the teacher) to another (student) network (Hinton et al., 2015; Duval et al., 2023; Zhang et al., 2019; Lopes et al., 2017; Wen et al., 2021; Caron et al., 2021; Zhai et al., 2023), and in other cases two models provide labels to a batch of samples which are optimized to have certain properties.

Pseudo-labeling techniques which are tolerant to label noise. The process of providing pseudo-labels is ultimately noisy. Many PL approaches are inspired by techniques known to be effective under label noise, or new mechanisms that are tolerant to label noise which are developed in this context. For example, the work of Angluin and Laird (1987) on random classification noise has inspired new learning algorithms in SSL for more than two decades now; e.g., Goldman and Zhou (2000) and Zhou and Li (2005). Similarly, other lines of work integrate recent developments on label noise, in a broader framework that uses pseudo-labels, and in particular in contexts that

[2]The third assumption is the *manifold assumption*, which states that high-dimensional data lies on a low(er)-dimensional manifold embedded within that space (Chapelle et al., 2006). This is equivalent to saying the dimensionality of the data can be meaningfully reduced.

utilize deep neural networks; e.g., Patrini et al. (2017); Guo et al. (2018); Wang et al. (2021); Xia et al. (2019); Han et al. (2018); Goldberger and Ben-Reuven (2017). Along these lines there is work that has been done to address labeling errors by human data annotators of varying expertise (and thus of varying label quality); e.g., Yan et al. (2014). Also, quite close to this context is the use of adversarial training as a mechanism that creates more robust models by using pseudo-labels; e.g., Miyato et al. (2019); Xie et al. (2020a).

Techniques that deal with label noise are deeply embedded to PL methods and one can find noise-tolerant mechanisms for PL approaches in all the categories shown in Figure 1. While the investigation of label noise and methods that mitigate label noise is fascinating in its own right, it is nevertheless outside the scope of the current survey. The interested reader may find more information in the references cited above and the references therein regarding the various approaches that are being followed towards noise mitigation. There are also excellent surveys on the topic; e.g., Song et al. (2023); Han et al. (2020); Frénay and Verleysen (2014) to name a few.

Novelty. While many surveys have offered comprehensive overviews of semi-supervised learning as a whole, they typically devote only limited space to pseudo-labeling, despite its centrality to modern practice. Pseudo-labeling is at once deceptively simple and remarkably influential: it forms the backbone of numerous state-of-the-art SSL algorithms (such as the *-Match family discussed in Section 3.1) and has inspired extensions into adjacent areas such as label-noise learning and weak supervision. By focusing exclusively on pseudo-labeling, our survey provides a systematic taxonomy, unifies disparate methodological threads, and clarifies conceptual underpinnings that broader SSL surveys necessarily treat only in passing. This narrower but deeper scope distinguishes our contribution and makes it complementary to existing work.

Outline. The following sections give an overview of PL approaches organized by supervision regime, mechanism of regularization used, and by model architectures (see Figure 1). In particular, in Section 2 we provide preliminaries so that we can clarify terms. Of particular importance are the notions of fuzzy sets, fuzzy partitions, and stochastic labels which allow us to define the central notion of this review, that of *pseudo-labels*. Once this common language is established, we proceed in Section 3 with a review of the methods that belong to semi-supervised learning. In Section 4 we discuss unsupervised methods, including methods of self-supervision. In Section 5 we discuss commonalities between the techniques that are presented in Sections 3 and 4. In Section 6 we discuss some directions for future work. Finally, we conclude our review in Section 7.

2 Preliminaries

Before we proceed with our presentation we give some definitions and background that can make the rest of the presentation easier and clearer. We will introduce an interpretation of pseudo-labeling as fuzzy partitions which provides a formal framework for understanding the methods in the remainder of the text.

2.1 Fuzzy Partitions

Fuzzy sets have the property that elements of the universe of discourse Ω belong to sets with some *degree of membership* that is quantified by some real number in the interval $[0, 1]$. For this reason a function $m : \Omega \to [0, 1]$ is used so that $m(\omega)$ expresses the *grade* (or, *extent*) to which an $\omega \in \Omega$ belongs to a particular set. The function $m = \mu_{\mathscr{A}}$ is called the *membership function* of the fuzzy set $\mathscr{A}$. Such a fuzzy set $\mathscr{A}$ is usually defined as $\mathscr{A} = (\Omega, \mu_{\mathscr{A}})$ and is denoted as $\mathscr{A} = \{\mu_{\mathscr{A}}(\omega)/\omega \mid \omega \in \Omega\}$.

Example 1 (Illustration of a Fuzzy Set). *Let $\mathscr{X}$ be the universe of discourse with just four elements; i.e., $\mathscr{X} = \{a, b, c, d\}$. Then, $\mathscr{S}_1 = \{0.4/a, 0.1/b, 0.0/c, 1.0/d\}$ is a* fuzzy set *on $\mathscr{X}$.*

The intention is to treat the "belongingness" numbers as probabilities that elements have for membership in particular sets. For pseudo-labeling we will assume that there is a predetermined number of fuzzy sets $\mathscr{S}_1, \ldots, \mathscr{S}_k$, for some $k \geq 2$ and these will correspond to the k hard labels that are available in a classification problem

(more below). Along these lines, an important notion is that of a *fuzzy partition*, which we will use to define pseudo-labeling precisely.

Definition 2.1 (Fuzzy Partition). Let $k \geq 2$. Let $\mathscr{I} = \{1, \ldots, k\}$. Let $\mathcal{Q} = (\mathcal{S}_1, \ldots, \mathcal{S}_k)$ be a finite family where for each $i \in \mathscr{I}$, $\mathcal{S}_i$ is a fuzzy set and $\mu_{\mathcal{S}_i}$ is its corresponding membership function. Then, $\mathcal{Q}$ is a *fuzzy partition* if and only if it holds that $(\forall x \in \mathscr{X}) \left[\sum_{i \in \mathscr{I}} \mu_{\mathcal{S}_i}(x) = 1\right]$.

Example 2 (Illustration of a Fuzzy Partition). *Let $\mathscr{I} = \{1, 2, 3\}$. Let $\mathscr{X} = \{a, b, c, d\}$ and consider the following fuzzy sets:*

$$\begin{cases} \mathcal{S}_1 &= \{\, 0.4/a,\ 0.1/b,\ 0.0/c,\ 1.0/d \,\} \\ \mathcal{S}_2 &= \{\, 0.3/a,\ 0.6/b,\ 0.6/c,\ 0.0/d \,\} \\ \mathcal{S}_3 &= \{\, 0.3/a,\ 0.3/b,\ 0.4/c,\ 0.0/d \,\} \end{cases}$$

The above family of fuzzy sets $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ together with the corresponding membership functions $\mu_{\mathcal{S}_1}, \mu_{\mathcal{S}_2}, \mu_{\mathcal{S}_3}$ whose values are shown in the equation on display, provides a fuzzy partition since, for every $x \in \mathscr{X}$, it holds that $\sum_{i \in \mathscr{I}} \mu_{\mathcal{S}_i}(x) = 1$.

More information on fuzzy sets is available at Zadeh (1965) and Ruspini (1969).

2.2 Basic Machine Learning Notation

We use $\mathscr{X}$ to denote the *instance space* (or, *sample space*) and $\mathscr{Y}$ the *label space*. We care about classification problems in which case it holds that $|\mathscr{Y}| = k$; that is there are k *labels* (*categories*). Instances will have *dimension* n. An *example* $(\mathrm{x}, y) \in \mathscr{X} \times \mathscr{Y}$ is a *labeled instance*. We denote an *unlabeled dataset* with $\mathscr{D}_{\mathrm{U}}$; e.g., $\mathscr{D}_{\mathrm{U}} = (\mathrm{x}_1, \ldots, \mathrm{x}_m)$ is an unlabeled dataset composed of m instances. If in addition, we have the true labels of all these instances, then we are working with a *labeled dataset* denoted with $\mathscr{D}_{\mathrm{L}}$; e.g., $\mathscr{D}_{\mathrm{L}} = ((\mathrm{x}_1, y_1), \ldots, (\mathrm{x}_m, y_m))$ is a labeled dataset with m examples.

Given access to datasets $\mathscr{D}_{\mathrm{U}}$ and $\mathscr{D}_{\mathrm{L}}$, the goal of semi-supervised learning is to develop a *model* f that approximates well some underlying function that maps the instance space $\mathscr{X}$ to the label space $\mathscr{Y}$. Examples of this behavior that we want to learn are contained in $\mathscr{D}_{\mathrm{L}}$ and additional instances are available in $\mathscr{D}_{\mathrm{U}}$. The model that we want to learn has trainable parameters θ which govern its predictions and for this reason we may write the model as f_θ so that this dependence on θ is explicit. The space of all possible functions f_θ that correspond to all the possible parameterizations θ is called the *model space* and is denoted by $\mathscr{F}$. With all the above definitions, in our context a *supervised learning problem* is specified by the tuple $(\mathscr{X}, \mathscr{Y}, \mathscr{F}, \mathscr{D}_{\mathrm{L}})$, whereas a *semi-supervised learning problem* is specified by the tuple $(\mathscr{X}, \mathscr{Y}, \mathscr{F}, \mathscr{D}_{\mathrm{L}}, \mathscr{D}_{\mathrm{U}})$.

We assume familiarity with artificial neural networks. However, the interested reader can find more information in a plethora of resources, including, e.g., (Shalev-Shwartz and Ben-David, 2014; Mitchell, 1997; James et al., 2023; Géron, 2022). To illustrate the relevance of Definition 2.1, we can see how it applies to the first deep learning application of pseudo-labeling: Lee (2013). Lee utilizes a neural network to perform image classification, which we will call f_θ. The neural network defines a fuzzy partition. For each classification problem there is an input space of images $\mathscr{X}$ and a number of target classes k. For $x \in \mathscr{X}$ we have $f_\theta(x) \in \Delta^k$, where Δ^k is the probability simplex of dimension k. Thus, each position of the vector $f_\theta(x)$ represents a class probability, or equivalently membership in a fuzzy set $\mathcal{S}_i$. Note that as training proceeds this network might be updated, but the network will continue to provide pseudo-supervision.

2.2.1 Deeper Discussion on Labels.

Below we provide more information on labels and pseudo-labels. In addition, Δ^k will continue to be the probability simplex of dimension k.

Definition 2.2 (Stochastic Labels). Let $|\mathscr{Y}| = k$. *Stochastic labels* are vectors $y \in \Delta^k$ where the positions of the vector defines a probability distribution over the classes $\mathscr{Y}_i \in \mathscr{Y}$ e.g., $y = (p_1, \ldots, p_k)$.

The idea of stochastic labels is very natural as the output of deep neural networks almost always has the form shown above. In order to make sure that the role of fuzzy sets and fuzzy partitions is clear to the reader, and what the relationship of these terms is to stochastic labels, below we revisit Example 2 and make some additional comments.

Remark 1 (Stochastic Labels from Fuzzy Partitions). *In Example 2, the fuzzy sets $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$ correspond to three different labels. By taking vertical cuts of these three fuzzy sets for each one of the four instances ($a, b, c, d \in \mathcal{X}$) one obtains the stochastic label associated with the respective instance. For example, the stochastic label of instance b is* $(0.1, 0.6, 0.3)$.

Definition 2.3 (Pseudo-Labels). Given a fuzzy partition $\mathcal{Q}$ over $\mathcal{X}$, pseudo-labels are vectors $y \in \Delta^k$ where each position of the vector defines the probability of membership in a unique fuzzy set in $\mathcal{Q}$ (corresponding to a label), but which were not specified as part of the learning problem. For some instance $x \in \mathcal{X}$ we have $y = \left(\mu_{\mathcal{S}_1}(x), \ldots, \mu_{\mathcal{S}_k}(x)\right)$. We write $y = \mathcal{Q}(x)$.

Using the notation introduced earlier, for some instance x, the pseudo-label takes the form $f_\theta(x)$. In other words, a pseudo-label is a stochastic label by definition. However, the crucial point is that a pseudo-label is obtained as a result of a predictive model $f_\theta(x)$ applied on the instance x, indicating the probability that x has belonging to different classes according to f_θ. Ideally, generated pseudo-labels are similar to ground-truth labels for the various instances $x \in \mathcal{X}$. Good pseudo-labels are similar to the ground truth to the extent that the ground truth is not significantly different from the labeled dataset $\mathcal{D}_\mathrm{L}$ (e.g., because of noise). Because each element of the pseudo-label vector indicates membership in a particular class, the pseudo-label can also be one-hot (e.g., see instance *d* in Example 2). The difference between the ground truth label and the pseudo-label in that case is that the latter was generated by the fuzzy partition using the instance.

Definition 2.4 (Pseudo-Labeling). Utilizing pseudo-labels generated by at least one fuzzy partition $\mathcal{Q}$ of the input space $\mathcal{X}$ inferred from only instances and ground truth labels to provide supervision during the training of a machine learning algorithm is called *pseudo-labeling*.

Pseudo-labeling is the process of generating/inferring stochastic labels. In our example the fuzzy class partition defined by f_θ is used to assign pseudo-labels to unseen instances which are then used to optimize the network further.

Definition 2.5 (Pseudo-Labeled Examples). Pseudo-labeled examples are tuples $(x, \mathcal{Q}(x))$.

The examples that are formed by the tuples of instances and pseudo-labels are the pseudo-examples which are used during retraining.

2.2.2 Data Augmentation. Data augmentation is a common technique across multiple branches of machine learning disciplines, broadly attempting to generate a set of derivative samples $\tilde{\mathrm{x}}$ from a single sample x by modifying the sample in some way. Augmenting samples can help normalize skewed sample class counts by artificially inflating a rare subclass (Shorten and Khoshgoftaar, 2019), and are used within many unsupervised learning methods to generate distinct samples that are known to be similar for consistency regularization (discussed in Section 4). These augmentations vary by implementation, but generally fall into one of the following categories (Shorten and Khoshgoftaar, 2019): *flipping* (flips the image 180 degrees along the horizontal or vertical axes), *color manipulation* (alters the color information of the image), *cropping* (randomly crops the image), *affine transformations* (rotates, translates, or shears the image), *noise injection* (add random noise to the image), *kernel filters* (applies e.g., a Sobel filter over the image), or random masking (blanks out a section of the image).

3 Semi-Supervised Regimes

The classic formulation of PL techniques as applied to deep neural networks is defined in Lee (2013), where PL is presented as a fine-tuning stage in addition to normal training. At a high level, the model f_θ is initially trained over samples from the labeled set $\mathscr{D}_\mathrm{L}$, and after a fixed number of epochs θ is frozen and pseudo-examples derived from f_θ and $\mathscr{D}_\mathrm{U}$ are added into the training steps, yielding an effective SSL technique. We use the symbol $f_{\theta'}$ to denote the model trained using the pseudo-examples.

More specifically, this formulation of PL derives labels for samples drawn from $\mathscr{D}_\mathrm{U}$ by "sharpening" the predictions of the partially-trained network, effectively reinforcing the partially-trained model's "best guess" for a particular sample. The loss function $\mathscr{L}$ being optimized has a supervised component ℓ_S and a nearly-identical unsupervised component ℓ_U; in every case cross-entropy loss is used to penalize the model's predictions against, respectively, the true labels, or the pseudo-labels obtained by the sharpening process:

$$\mathscr{L} = \underbrace{\ell_\mathrm{S}(y, f(\mathrm{x}_\mathrm{S}))}_{\text{supervised}} + \alpha(t)\underbrace{\ell_\mathrm{U}(f_\theta(\mathrm{x}_\mathrm{U}), f_{\theta'}(\mathrm{x}_\mathrm{U}))}_{\text{unsupervised}} \tag{1}$$

Crucially, the unsupervised segment is moderated by $\alpha(t)$: an epoch-determined annealment parameter. This starts at zero and then ramps linearly up to a ceiling value. According to Lee (2013), careful consideration is required in choosing the scheduling of α, as setting it too high will disturb training of labeled data and too low will not have an impact on the training process at all.

Thus the basic foundation upon which most fine-tuning PL methods is built. The remainder of this section presents modifications and improvements to this formulation. It should be clear from the previous sections how this fits into our definition for pseudo-labeling: the unsupervised portion of the loss is computed through the use of an inferred fuzzy partition (defined by the weights of the model trained so far). In the case of hard pseudo-labels the membership probabilities are one-hot, and in the case of soft pseudo-labels they may not be. Lee (2013) evaluates adding a denoising variational autoencoder in order to simplify the representations of the samples in both datasets $\mathscr{D}_\mathrm{L}$ and $\mathscr{D}_\mathrm{U}$ to boost the performance of the PL steps. For example, assigning sharpened labels to samples drawn from $\mathscr{D}_\mathrm{U}$ can make the training objective unstable when pseudo-labeling some unlabeled samples changes dramatically from one weight update to the next (Arazo et al., 2020). Consequently, one of the primary focuses of work in this direction is how to best choose the pseudo-labels or examples such that training progresses advantageously.

In this section we will provide an overview of pseudo-labeling methods in computer vision for semi-supervised learning. We will discuss methods based on scheduling unlabeled instances (Section 3.1), methods based on performing well under weak supervision (Section 3.2), methods of consistency regularization (Section 3.3), and approaches that use multiple models (Section 3.4).

3.1 Sample Scheduling

An inherent prerequisite to solving the problem of generating pseudo-labels for samples is determining for *which* samples to generate pseudo-labels. The precise sample schedule can greatly impact the performance of the final pseudo-labeling algorithm (Arazo et al., 2020). Across the literature, a variety of techniques have been proposed as improvements to Lee (2013) by altering the manner in which unlabeled instances are selected and how pseudo-labels are assigned to them.

3.1.1 Random Selection. The simplest way to choose unlabeled samples to label is to sample them randomly from the unlabeled set. Pseudo-labeling techniques are vulnerable to *confirmation bias*, where initially-wrong predictions are reinforced by the PL process (Arazo et al., 2020). One solution is to reduce the confidence of the network predictions overall (Arazo et al., 2020). This is a feature of *MixMatch* (Berthelot et al., 2019b), a hybrid PL technique combining several dominant methods within the SSL space with the goal of preventing

overconfident predictions. MixMatch consists of several steps: a *data augmentation* step, a *label guessing* step (the PL component), and a *sample mixing* step. First, a batch of samples is drawn from both the labeled set $\mathcal{X} \subset \mathcal{D}_{\mathrm{L}}$ and the unlabeled set $\mathcal{U} \subset \mathcal{D}_{\mathrm{U}}$. Samples from $\mathcal{X}$ are then replaced with stochastically-augmented versions to form $\hat{\mathcal{X}}$, and samples from $\mathcal{U}$ are replaced by K augmentations of the each sample to form $\hat{\mathcal{U}}$. Samples from $\mathcal{U}$ are pseudo-labeled based on the sharpened average of the model's predictions for all K augmentations of each sample. Finally, these samples are then shuffled and mixed using MixUp (Berthelot et al., 2019b; Zhang et al., 2018) to yield two mixed distributions $\mathcal{X}', \mathcal{U}'$ containing labeled and unlabeled samples.

ReMixMatch (Berthelot et al., 2019a) was proposed with two additional concepts on top of MixMatch: *distribution alignment* and *augmentation anchoring*, and a modification of the loss function to improve stability. Distribution anchoring is the scaling of each prediction's output probabilities element-wise by the ratio of the output to the average of the outputs over a number of previous batches—replacing the sharpening function in MixMatch and serving as an EM term. The other contribution was augmentation anchoring, which replaced MixMatch's one-step stochastic augmentation of a true sample with a two step process: first weakly augmenting a sample, and then using that sample-label pair as a base for a series of K strong augmentations (using their proposed CTAugment, a version of AutoAugment (Cubuk et al., 2018)). This functions as a consistency regularization technique, and is similar to methods discussed in Section 4 such as SimCLR (Chen et al., 2020). Additionally, ReMixMatch's loss function incorporates two additional targets: a *pre-MixUp unlabeled cross-entropy loss* component (comparing unmixed unlabeled samples with their guessed labels) and a *rotation loss* component (rotating a sample randomly in 90 deg increments and predicting the rotation of the sample). These changes yield a significant improvement in the label efficiency of the model, with the authors showing a $5 - 16\times$ reduction in required labels for the same accuracy.

Limitations. Sample scheduling approaches, such as the *-Match methods, rely on a carefully chosen recipe of data augmentation strategies. These strategies are essential to effective consistency regularization, and to some extent entropy minimization as well. For some well-known datasets there exist popular data augmentation policies like those of AutoAugment Cubuk et al. (2018) on CIFAR10 and CIFAR100, however for novel domains it may require extensive experimentation to construct such a policy. Since constructing an effective strong data augmentation strategy is a prerequisite to applying the above semi-supervised methods, this may pose a limitation to applying them to some data domains.

With random selection it is assumed that the unlabeled data follow the same distribution as the labeled data. This assumption may not always be realistic, thus leading to poor generalization ability of the final model in challenging data scenarios.

3.1.2 Confidence-Based Selection. An alternative approach to random sampling is metric-based sampling, where the sample to pseudo-label is selected from the unlabeled pool $\mathcal{D}_{\mathrm{U}}$ via a non-random metric such as confidence. One such method is *FixMatch* (Sohn et al., 2020); a simplification of the ReMixMatch process. FixMatch again has two loss terms: labeled and unlabeled. The labeled loss is cross-entropy computed over labels given weakly augmented (flipped and cropped) versions of supervised examples. The unlabeled loss is computed only for confident predictions on weakly augmented versions of the images from the unlabeled set. This is equivalent to filtering unlabeled data based on confidence. Pseudo-labels are again computed from the weakly augmented versions of unlabeled samples, but in FixMatch, hard labels are assigned. FixMatch includes the labeled data (without labels) in the unlabeled set during training.

While most methods use confidence as the metric by which to select, *Time-Consistent Self-Supervision for Semi-Supervised Learning* (TCSSL) (Zhou et al., 2020) uses a different metric. For each gradient update step, TCSSL maintains a "time consistency" metric which is the exponential moving average of the divergence of the discrete output distribution of the model for each sample. They show this metric is positively correlated with accuracy,

especially at later epochs, and that the more commonly used confidence is not. In addition to PL, this approach includes jointly in its objective representation learning losses.

Limitations. Selecting pseudo-labels based on a model's confidence may not be effective if the model's uncertainty is poorly calibrated. In general it is known that training of neural networks via gradient descent methods does not necessarily produce calibrated uncertainty estimates.

3.1.3 Curriculum Learning. One of the most natural solutions to choosing a suitable set of examples to pseudo-label is *curriculum learning* (CL) (Bengio et al., 2009). In this paradigm, the model is first allowed to learn on "easy" instances of a class or concept and then trained on progressively "harder" instances. Instances which are "easy" are ones which will be given correct pseudo-labels (with high probability), and thus we have extended our labeled set reliably. Through training in this way the hope is that harder examples become easier and we can continue to expand our labeled set. Unfortunately, designing a curriculum is not easy and often requires handcrafting a heuristic for determining what is easy and what is hard.

One such CL approach is (Cascante-Bonilla et al., 2020). Functionally, it is a self-training framework where within each epoch unlabeled data is sorted based on the confidence of prediction, and only the top $k\%$ most confident predictions are given pseudo-labels at each iteration of self training until all labels are exhausted. Curriculum learning is also the foundation of FlexMatch (Zhang et al., 2021), an extension to the *-Match family of methods (see Section 3.1.1). FlexMatch extends FixMatch's pseudo-label thresholding by introducing dynamically-selected thresholds per-class (dubbed *Curriculum Pseudo-Labeling* or CPL), and these are scaled to the "difficulty" of learning a particular class' label. This is achieved through two assumptions: the first that the dataset is class-balanced and the second that the difficulty of learning a particular class is indicated by the number of instances with a prediction confidence above a given threshold τ. During the training process, τ is adjusted down for "hard" classes to encourage learning on more pseudo-labels, and up for "easy" classes in order to avoid confirmation bias (Zhang et al., 2021). This improves SSL performance without much additional cost, as there are no extra forward passes and no extra learnable parameters (Zhang et al., 2021).

Limitations. As we discussed previously, neural networks do not necessarily produce calibrated uncertainty estimates and choosing an effective data augmentation strategy is not always trivial. Both of these limitations apply to the above methods with respect to choosing a curriculum for pseudo-labeling. Additional effort may have to be invested in calibrating model outputs or developing an effective data augmentation strategy to apply methods of curriculum-learning-based pseudo-labeling.

3.2 Weak Supervision

Subtly different from metric-based selection of unlabeled examples, we can also imagine assigning a label based on a weak supervision such as using a k-nearest-neighbors (kNN) classifier for PL. Such a classifier differs from the metrics in the previous section as they only provide a label rather than a continuous score. This updated task brings pseudo-labeling to *weak supervision*, a class of problem similar to SSL where the labels in $\mathscr{D}_{\mathrm{L}}$ are treated as inaccurate, and are updated along with the pseudo-labels at training time.

An early approach in this space was that of Zhu and Ghahramani (2002), which used kNN to assign a label based on its geodesic distance to other samples in the embedding space. This approach was extended to deep neural networks in Iscen et al. (2019). Under the deep learning regime, a CNN is trained on the labeled data and the representation output by the penultimate layer is used as an approximation of the underlying data manifold. Then, an iterative algorithm assigns soft pseudo-labels to the unlabeled data based on density and geodesic distance to labeled examples. The CNN is retrained on these new targets in addition to the original labels. This process is iterated until some stopping criterion is satisfied. Shi et al. (2018) improves upon this with a regularization they call "Min-Max Feature," a loss which encourages all examples of one class assignment to

be embedded far from those of other classes and examples within an assignment to be drawn together. During model optimization, this regularization is performed during a separate step subsequent to the optimization of the cross-entropy loss over the union of the labeled and pseudo-labeled sets. At the beginning of each self-training iteration hard pseudo-labels are assigned as the argmax of the output of the CNN.

Iterative Label Improvement (ILI) (Haase-Schütz et al., 2020) recognizes the potential inaccuracies in supervised dataset labels, emphasizing the inherent noise in pseudo-labels. ILI employs a self-training method where the model, post-initial training, may adjust any sample's label based on confidence levels, without calibration. Additionally, ILI extends through *Iterative Cross Learning* (ICL) (Yuan et al., 2018), where models trained on dataset partitions assign pseudo-labels to other segments. *Uncertainty-Aware Pseudo-Label Selection* (UPS) (Rizve et al., 2021) seeks to achieve good weak supervision by training a calibrated network for confidence filtering on the unlabeled set. Pseudo-labels generated by this method are hard, and only computed after each iteration of self-training.

Continuing in the theme of directly estimating the label noise, Tong Xiao et al. (2015) add a probabilistic model which predicts the clean label for an instance. This approach splits noise in labels into two categories: "confusing noise" which are due to poor image/label matching (though still conditional on the label), and "pure random noise" which is unconditional and simply wrong. The model represents the image/noisy label relationship by treating true label/noise types as latent variables. This latent variable, z, is a one-hot encoded 3-element vector with the following properties:

(1) $\mathrm{z}_1 = 1$: Label is noise free,
(2) $\mathrm{z}_2 = 1$: Label is unconditionally noisy (uniform random distribution),
(3) $\mathrm{z}_3 = 1$: Label is conditionally noisy (moderated by learnable matrix C).

Two CNNs are then trained to estimate the label and noise type respectively: $p(\mathrm{y} \mid \mathrm{x})$ and $p(\mathrm{z} \mid \mathrm{x})$. The CNNs are pretrained using strongly supervised data, and then noisy data is mixed into clean data on an annealing schedule. In benchmarks, this approach yielded a 78.24% test accuracy, a modest improvement over treating the noisy dataset as unlabeled and using pseudo-labeling (73.04% test accuracy) and treating the noisy labels as ground truth (75.30% test accuracy).

More recently, PENCIL (Yi et al., 2022) handles uncertainty by modeling the label for an image as a distribution among all possible labels and jointly updates the label distribution and network parameters during training. PENCIL extends DLDL (Gao et al., 2017) by using the inverse-KL divergence ($\mathrm{KL}(f(\mathrm{x}_i;\theta)\|\mathrm{y}_i^d)$ rather than $\mathrm{KL}(\mathrm{y}_i^d\|f(\mathrm{x}_i;\theta))$) to moderate the label noise learning, as KL divergence is not symmetric and this swap gives a better gradient (*i.e.,* not close to zero) when the label is incorrect. Additionally, two regularization terms are added to control the learning: a *compatibility loss* $\mathscr{L}_o$ and an *entropy loss* $\mathscr{L}_e$. The compatibility loss term ensures that the noisy labels and estimated labels don't drift too far apart—given that the noisy labels are mostly correct—and is a standard cross-entropy between the two terms. The entropy loss $\mathscr{L}_e$ serves to ensure that the network does not just directly learn the noised weights, but rather encourages a one-hot distribution. These one-hot labels are correct for classification problems and keeps training from stalling. The final loss is a sum of the classification loss and hyperparameter-moderated compatibility/entropy losses. Similar to other pseudo-labeling techniques, PENCIL pre-trains the backbone network assuming that the existing labeling is correct. Subsequently, the compatibility and entropy losses are brought in for several epochs before a final fine-tuning is done over the learned new labels. This approach makes PENCIL useful for both semi-supervised and weakly supervised regimes.

Another approach in this space is *instance dependent noise* (IDN) by Wang et al. (2022), which directly models label noise by relating noisy labels to instances. Here, a DNN classifier is trained to identify an estimate of the probability $P(\text{confusing} \mid x)$, and this classifier is optimized using a variation of the expectation-maximization

algorithm. This is shown to be slightly more effective than PENCIL, with the added benefit that the confusing instance classifier gives an explainable view of which instances are confusing.

An additional strategy for label selection is *Class Introspection* (Kage and Andreadis, 2021), which uses an explainability method over a classifier in order to embed the inputs into a space separated by their learned explanations. This space is then hierarchically clustered in order generate a new (pseudo-) labeling, with the intuition that points which were included in a single class but are explained differently may belong in a separate, new labeling.

Limitations. Necessarily using weak supervision to assign pseudo-labels requires a source of that weak supervision, and are limited by the quality of that weak supervision. Approaches like label propagation, LabelPropPL, and MMF require a feature space over which labels can be propagated correctly. Approaches like ILI and UPS which require good uncertainty estimates incur additional training cost. Popular ways to train uncertainty-aware networks include Bayesian Neural Networks (Graves, 2011; Blundell et al., 2015; Louizos and Welling, 2016), Deep Ensembles (Lakshminarayanan et al., 2017), and MC Dropout (Gal and Ghahramani, 2016) all come at the expense of extra computation. Finally, methods which rely on auxiliary trained classifiers like IDN incur computational cost and hyperparameter tuning burden by adding another trainable component.

3.2.1 Open-Set Recognition. Open-set recognition (OSR) and pseudo-labeling, while distinct, share a connection in weakly-supervised tasks. OSR aims to identify known classes while detecting and rejecting unknowns (out-of-distribution instances) (Barcina-Blanco et al., 2024), a challenge that becomes more complex without strong labels. In the context of weakly-supervised OSR, pseudo-labeling can be employed to create initial, albeit noisy, labels for the known classes in unlabeled data. These pseudo-labels help define boundaries for known classes, which is crucial for OSR to distinguish them from unknowns. Thus, pseudo-labeling provides a form of implicit supervision, enabling OSR models to learn and differentiate classes even with limited or absent explicit labels.

An approach in this space is SeFOSS (Wallin et al., 2023), which aims to learn from all data (in and out of distribution) in $\mathscr{D}_{\mathrm{U}}$ and uses a consistency regularization objective to pseudo-label inliers and outliers for the dataset. SeFOSS utilizes the *free energy score* of the dataset (as defined in (Liu et al., 2021)) rather than a softmax objective to filter $x \in \mathscr{D}_{\mathrm{U}}$ into three buckets: *pseudo-inliers*, *uncertain data*, and *pseudo-outliers*. Similar to FixMatch (Sohn et al., 2020), these are sorted by thresholding, however because free energy is based on the raw logits rather than a normalized probability distribution SeFOSS delegates these (scalar) thresholds to be assigned as hyperparameters.

Free energy is not the only metric used for in-/out-of-distribution assignment. ProSub (Wallin et al., 2024) uses the angle θ between clusters in the penultimate layer activations of a neural network to assign instances to the in- or out-distribution, and avoids having to manually set thresholds by estimating the probability of being in-distribution as a Beta distribution. This Beta distribution is computed using a variant of the standard EM algorithm, and the resulting ID/OOD prediction is used to weight the instance in the loss function. ProSub finds that this achieved SOTA results on benchmarks, showing the efficacy of this method.

While weak supervision is a practical way to incorporate noisy labels into the training process, it is not without its drawbacks. A major challenge is managing dataset noise (ensuring datasets are not too noisy to learn meaningfully from), and ensuring that overconfident predictions do not amplify biases within the dataset itself. Special care needs to be taken when evaluating whether a weakly supervised method is applicable or if a fully unsupervised approach is more appropriate.

Outlier exposure—ensuring the model is presented with a variety of out-of-dataset instances during training time—is another challenge within OSR as it is difficult to collate a diverse set of outliers. One solution is Wasserstein Distribution-Agnostic Outlier Exposure (W-DOE), which includes a noise term $\mathrm{P}^{(k)}$ into the classic MLP weights-and-nonlinearity formulation of $\sigma(\mathrm{W}^{(k)}\mathrm{x}^{(k)})$, yielding a noised variant $\sigma((\mathrm{W}^{(k)} + \mathrm{P}^{(k)})\mathrm{x}^{(k)})$ which is shown to

be equivalent to perturbing the inputs x of the model Wang et al. (2025). This noising of the model is used to generate data outside of the dataset distribution (bounded by a parameter ρ indicating maximum Wasserstein distance), and these outliers are used to train the model. This resulting system outperforms the state-of-the-art at discriminating OOD samples.

Limitations. As discussed above, a specific challenge with open-set recognition methods is ensuring that the out-of-distribution samples presented during training align with the samples encountered in the real world. While the methods discussed in this section are powerful tools, they need to be benchmarked against not just the reference datasets but also samples drawn from the model's deployment environment—a potentially intractable problem depending on the eventual use-case.

3.3 Consistency Regularization

A common problem with models trained on few labeled examples is that of overfitting. There are two representative techniques which inject consistency regularization in a way which specifically addresses the case where pairs of instances that differ only slightly receive differing model predictions. These are Unsupervised Data Augmentation (UDA) by Xie et al. (2020a) and Virtual Adversarial Training (VAT) by Miyato et al. (2019). It is possible to apply these methods without any supervision at all, and in that case they are still pseudo-labeling, but they would be unsupervised rather than semi-supervised. UDA is even explicitly called *unsupervised* data augmentation, but because historically these two works are specifically for semi-supervised applications we include them in this section rather than the next.

Unsupervised data augmentation (Xie et al., 2020a) is a method for enforcing consistency regularization by penalizing a model when it predicts differently on an augmented instance. UDA employs a consistency loss between unaugmented and an augmented version of a pseudo-labeled instance. If the two predictions differ, then the pseudo-label on the unaugmented instance is assumed to be correct and the model is trained to reproduce that label for the augmented instance. In this way the model learns to be invariant to the augmentation transformations. For its simplicity UDA is remarkably effective, and it bears strong resemblance to many of the CR methods of self-supervised learning in the next section. UDA does, however, require an effective data augmentation strategy and is typically achieved with a strategy that is known to be effective already on a particular dataset.

Virtual adversarial training (VAT) is a CR technique which augments the training set by creating new instances that result from original samples with small adversarial perturbations applied to them. A consistency loss is applied between the new adversarial instance and the original instance. If the model's prediction differs then the pseudo-label for the original instance is assumed to be correct. The goal of this process is to increase the robustness and generalization of the final model. Rather than enforcing consistency by encouraging the model to be invariant to combinations of pre-defined transformations, VAT makes the assumption that examples within a ball of small radius around each instance should have the same label.

In both of the above cases the pseudo-label of an augmented instances is inferred by its close proximity to the original sample. This distance is either a p-norm in the case of VAT or a semantic notion of distance as in UDA. The fuzzy partition is thus constituted of fuzzy sets which are the union of spheres of a chosen radius around points of the same class, or which are sets of augmented versions of an instance. In the case of VAT the labels that are assigned are one-hot, but this is a subset of the admissible fuzzy partitions. Adversarial training is a special case of consistency regularization in which the consistency is enforced locally for each instance. Adversarial training is an effective method of consistency regularization, but it incurs the additional cost of a gradient update to the input sample.

Limitations. The benefit of consistency regularization is dependent on the ability to define transformations which do not effect the ground-truth label of an instance but to which the regularized classifier is not already

invariant. For image classification on benchmark tasks there exist out-of-the-box solutions, but for specialized domains or model architectures it may take non-trivial work to identify and leverage such transformations. Furthermore these transformations must meaningfully change the classification boundary. Using VAT with some small radius, for example, is broadly applicable but may not change the classification boundary much.

3.4 Multi-Model Approaches

In multi-model approaches more than one network is used for training one singular model, combining the strengths of the individual component networks. Meta Pseudo Label (MPL) by Pham et al. (2021), discussed in Section 3.4.1 of joint-optimization selection, is an example where a teacher model generates labels and a student model processes labeled data and jointly optimizes both.

Noisy Student Training (Xie et al., 2020b) is an effective technique in this category where a student model is initially trained on labeled and pseudo-labeled data. This student then becomes the teacher, and a new, possibly larger, "noised" student model takes its place, undergoing augmentation and stochastic depth adjustments. Training involves using a mix of supervised and pseudo-labeled data, with the teacher model filtering pseudo-labeled examples based on confidence. The process typically involves a few iterations of self-training, after which the final student model is selected. This approach differs from model distillation in that the student model is often larger, subjected to noise, and utilizes unlabeled data, which is not the case in distillation. The addition of noise is believed to make the student model mimic a larger ensemble's behavior.

However, the student-teacher model is not the only multi-model architecture that relies on pseudo-labels for SSL. A representative technique in this area is *co-training* (Blum and Mitchell, 1998) in which two models are trained on disjoint feature sets which are both sufficient for classification. In co-training, pseudo-labels are iteratively incorporated into the training set just as with self-training, except labels added to one model's training set are generated by the other model. The models need not have the same number of parameters, the same architecture, or even be of the same type. In Qiao et al. (2018) the authors present a study in which two similar deep neural networks are used. They call this *Deep Co-Training*. One key problem in co-training is that of constructing the views when they are not available (Sun et al., 2011; Wang et al., 2008). It is relatively uncommon to have a dataset in which there are two independent and sufficient views for classification. Furthermore, it is possible that the views we have are independent, but not sufficient. Multiple works attempt to address the issue of insufficiency. Wang and Zhou (2013) shows that this is still an acceptable setting in which to perform co-training provided that the views are "diverse."

To address the problem of the lack of independent views, some algorithms propose using a diverse group of classifiers on the same view instead of constructing or otherwise acquiring multiple sufficient and independent views (Zhou and Goldman, 2004; Hady and Schwenker, 2008). These approaches are ensembles where each classifier issues a vote on the class of unlabeled samples, and the majority vote is assigned as the pseudo-label and the instance is added to the training set. In this setting, it is reasonable to assume that the single view is sufficient, as in most supervised learning scenarios we are simply attempting to train the best classifier we can with the data we are given. Independence is far more dubious, as we rely on the diversity of the ensemble to ensure our classifiers do not all fail for the same reasons on points which they classify incorrectly.

In the same vein, *Tri-Training* employs three classifiers in order to democratically select pseudo-labels for $\mathscr{D}_{\mathrm{U}}$ after being trained on $\mathscr{D}_{\mathrm{L}}$ (Zhou and Li, 2005). While this is a simple solution, it comes with a 3× increase in compute cost. An improvement is *Robust Semi-Supervised Ensemble Learning* (Yan et al., 2016), which formulates the PL process as an ensemble of low-cost supervisors whose predictions are aggregated together with a weighted SVM-based solver.

Multi-Head Co-Training (MHCT) (Chen et al., 2022) uses a shared network block to create an initial representation which is then evaluated with multiple classification heads (see Figure 2). With labeled data, these

classification heads (and the shared block) are then all updated as normal. However, when an unlabeled sample is selected, the most confident pseudo-label is assigned based on a consensus between a majority of the heads. For better generalization MHCT uses data augmentation.

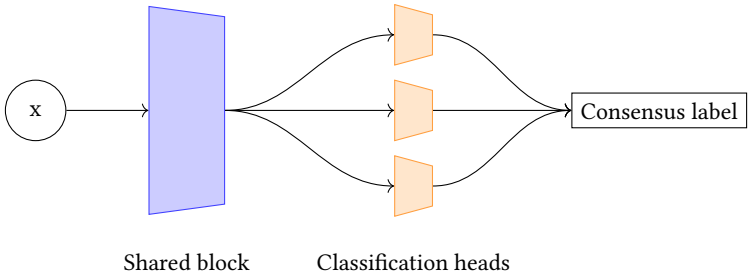

Fig. 2. A simplified diagram of Multi-Head Co-Training's architecture. Adapted from Chen et al. (2022).

Limitations. Methods that require training multiple models have the obvious drawback of the additional time and memory resources required to train those models. Often these methods are employing iterative re-training to learn from the pseudo-labels they produce, so the computational cost of that re-training is multiplied by the number of models used. Additionally, multi-model methods require that each of the models has captured some independent information or pattern from the data. If all of the models trained are more or less identical, then it is not useful at all to have trained multiple models. In cases where single model approaches are already very performant it can be challenging to find or train multiple models with competitive performance that capture independent information.

3.4.1 Joint-Optimization Selection. Unlike other methods of pseudo-labeling, *Meta Pseudo Label* (MPL) by Pham et al. (2021) treats pseudo-label assignment as an optimization problem, with the specific assignment of pseudo-labels jointly optimized with the model itself. The framework consists of two models, a student and a teacher, which share an architecture. The teacher receives unlabeled examples and assigns pseudo-labels to them. The student receives the example pseudo-label pair, and performs a gradient update. The student then is evaluated on a labeled sample. The teacher receives the student performance on the labeled sample, and performs a weight update to make better pseudo-labels. Note that the student is *only* trained on pseudo-labels and the teacher is *only* trained on the student's response on labeled data. In this way, the teacher learns an approximation of the optimal PL policy with respect to optimizing the student performance on the labeled set, and the student learns to approximate the PL function described by the teacher.

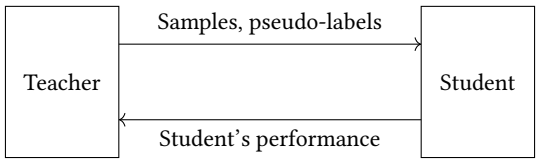

Fig. 3. A figure describing Meta Pseudo Label's overall architecture. Adapted from Pham et al. (2021).

A simplified overview of the MPL method is shown in Figure 3. First, the teacher f_{θ_T} is fixed and the student is trying to learn a better model by aligning its predictions to pseudo-labeled batches $(X_u, f_{\theta_T}(X_u))$. On a pseudo-labeled batch, the student optimizes its parameters using:

$$\theta_S^{\text{PL}} \in \text{argmin}_{\theta_S} \ell\big(\text{argmax}_i f_{\theta_T}(X_u)_i, f_{\theta_S}(X_u)\big)\,. \tag{2}$$

The hope is that updated student parameters θ_S^{PL} will behave well on the labeled data $L = (X_L, Y_L) = \{(x_i, y_i)\}_{i=1}^m$; i.e., the loss $\mathscr{L}_L\left(\theta_S^{\text{PL}}(\theta_T)\right) = \ell\left(Y_L, f_{\theta_S^{\text{PL}}}(X_L)\right)$. $\mathscr{L}\left(\theta_S^{\text{PL}}(\theta_T)\right)$ is defined such that the dependence between θ_S and θ_T is explicit. The loss that the student suffers on $\mathscr{L}_L\left(\theta_S^{\text{PL}}(\theta_T)\right)$ is a function of θ_T. Exploiting this dependence between θ_S and θ_T. and making a similar notational convention for the unlabeled batch $\mathscr{L}_u(\theta_T, \theta_S) = \ell\left(\text{argmax}_i f_{\theta_T}(X_u)_i, f_{\theta_S}(X_u)\right)$, then one can further optimize θ_T as a function of the performance of θ_S:

$$\begin{aligned} &\min_{\theta_T} \mathscr{L}_L\left(\theta_S^{\text{PL}}(\theta_T)\right)\,, \\ &\quad \textit{where} \\ &\quad \theta_S^{\text{PL}}(\theta_T) \in \text{argmin}_{\theta_S} \mathscr{L}_u(\theta_T, \theta_S)\,. \end{aligned} \tag{3}$$

However, because the dependency between θ_T and θ_S is complicated, a practical approximation is obtained via $\theta_S^{\text{PL}} \approx \theta_S - \eta_S \cdot \nabla_{\theta_S} \mathscr{L}_u(\theta_T, \theta_S)$ which leads to the practical teacher objective:

$$\min_{\theta_T} \mathscr{L}_L\left(\theta_S - \eta_S \cdot \nabla_{\theta_S} \mathscr{L}_u(\theta_T, \theta_S)\right)\,. \tag{4}$$

Whereas MPL optimizes a labeler to provide good labels to unlabeled points, Meta-Semi (Wang et al., 2020) optimizes which pseudo-labels to use for training. Particularly, if inclusion of a pseudo-label causes a decrease in loss for the student on a labeled batch then the pseudo-labeled example is included for training and otherwise it is omitted. Similarly, Bi-Level Optimization for Pseudo-Labeling Based Semi-Supervised Learning (BLOPL) (Heidari and Guo, 2025) considers pseudo-labels as latent variables of which the learner's weights are a function. Rather than optimizing the presence or absence of pseudo-labels BLOPL optimizes soft pseudo-labels which are always included to result in a good learned model. Reinforcement Learning Guided Semi-Supervised Learning (Heidari et al., 2024) poses this optimization of pseudo labels as a reinforcement learning problem and learns a policy to provide good soft pseudo-labels to the learner.

Meta Co-Training (MCT) (Rothenberger and Diochnos, 2023) combines the ideas of Meta Pseudo Labels (Pham et al., 2021) and Co-Training (Blum and Mitchell, 1998). Meta Co-Training introduces a novel bi-level optimization over multiple views to determine optimal assignment of pseudo labels for co-training. At the lower level of the optimization the student parameters θ_S are optimized as a function of the teacher parameters θ_T:

$$\mathscr{L}_u(\theta_T, \theta_S) = \ell\left(\text{argmax}_\xi f_{\theta_T}(X_u)|_\xi, f_{\theta_S}(X_u)\right) \tag{5}$$

$$\theta_S' = \theta_S^{\text{PL}}(\theta_T) \in \text{argmin}_{\theta_S} \mathscr{L}_u(\theta_T, \theta_S)\,. \tag{6}$$

At the upper level (Equation 8) the teacher parameters are optimized to improve the student:

$$\mathscr{L}_L\left(\theta_S'\right) = \ell\left(Y_L, f_{\theta_S'}(X_L)\right) \tag{7}$$

$$\theta_T' = \text{argmin}_{\theta_T} \mathscr{L}_L\left(\theta_S'\right)\,. \tag{8}$$

All together the objective of MCT from the perspective of the current view model as the teacher is:

$$\min_{\theta_T} \mathscr{L}_u(\theta_T, \theta_S) + \mathscr{L}_L\left(\theta_S'\right)\,. \tag{9}$$

The full objective of MCT is then:

$$\min_{\theta_1,\theta_2} \mathscr{L}_u(\theta_1,\theta_2) + \mathscr{L}_L\left(\theta_2'\right) + \mathscr{L}_u(\theta_2,\theta_1) + \mathscr{L}_L\left(\theta_1'\right) \tag{10}$$

In this formulation each view is utilized by a single model which serves as both the student for that view and the teacher for the other view. The student is optimized to replicate the pseudo-labels assigned by its teacher given the corresponding instance as input. The teacher is optimized to produce pseudo-labels that improve student performance on a held-out labeled set. This formulation is a natural extension of the ideas of Meta Pseudo Labels to two views. They further propose using pre-trained models to construct two views from a single instance in the dataset. This method of view construction compresses the input significantly in the case of image classification which mitigates the impact of the expensive computational nature of multi-model semi-supervised approaches, though it requires the existence of such models.

Limitations. Each of Pham et al. (2021); Rothenberger and Diochnos (2023); Heidari and Guo (2025); Heidari et al. (2024); Wang et al. (2020) pose a bi-level optimization in which the way in which pseudo-labels are provided to the learner is optimized to create a performant learner. Any approach based on a bi-level optimization has the potential to be expensive to compute. In the case of MPL, it takes a million optimization steps on the CIFAR-10 dataset to produce results only marginally better than Unsupervised Data Augmentation (Xie et al., 2020a). One of the major drawbacks of MPL and similar methods are their long training times.

4 Unsupervised and Self-Supervised Regimes

While pseudo-labeling is usually presented in the context of semi-supervised training regimes, there are very close analogues to PL in the domain of unsupervised learning. Specifically, unsupervised consistency regularization approaches show significant similarities with pseudo-labeling approaches as they tend to assign labels to points even if these labels have no relation to any ground truth. For example, EsViT (Li et al., 2022), DINO (Caron et al., 2021), BYOL (Grill et al., 2020), and SwAV (Caron et al., 2020) all use some form of label assignment as a representation learning *pretext task*, i.e., a task which prepares a model for transfer learning to a downstream task (Jing and Tian, 2021). Within this section, we will discuss descriminative self-supervised learning which operate as CR methods (Section 4.1), and response-based knowledge distillation (Section 4.2) both of which are forms of pseudo-labeling.

4.1 Consistency Regularization

It is perhaps surprising to consider pseudo-labeling as a definition for unsupervised learning settings, where traditional, fixed class labels are not available. However, such probability vectors are used in the training of particularly descriminative self-supervised learning. These approaches such as DINO (Caron et al., 2021) and are incorporated into the loss function of SWaV (Caron et al., 2020), EsViT (Li et al., 2022), and Deep Cluster (Caron et al., 2018) and their effectiveness is explained as a form of consistency regularization.

These approaches all result in assigning instances to a particular cluster or class which has no defined meaning, but is rather inferred from the data in order to accomplish the minimization of the self-supervised loss. The goal of the clustering or classification is clear: to place images which share an augmentation relationship into the same cluster or class, and optionally a different class from all other images. The partition of the latent space however is inferred, and it is also fuzzy as membership probabilities may not be one-hot. Thus, these techniques clearly satisfy our definition.

All these frameworks are "unsupervised" in the sense that they do not rely on ground-truth labels, but they truly belong to the category of SSL as they are trained on pretext tasks and are typically applied to semi-supervised tasks. A common adaptation (and indeed the standard metric by which representational performance is judged) is training a linear layer on top of the learned representations (Kolesnikov et al., 2019; Chen et al., 2020), with

the intent of performing well in settings where few labels are available. A common theme in solutions in this space is stochastically augmenting a pair of samples and ensuring that the learned representation of those samples is similar to each other and different to all other samples in the batch, for example as in Figure 4; see, e.g., Caron et al. (2020); Li et al. (2022); Caron et al. (2018); Xie et al. (2020a). In this sense, these are self-supervised learning techniques that technically use pseudo-labels. Such self-supervised learning approaches are referred to as "discriminative" as they are designed to discriminate between different ground truth signals by assigning different (pseudo) labels to them. Because they do not predict labels that correspond to any specific important function assignment, they are functionally indistinguishable from techniques like SimCLR (Chen et al., 2020), and BYOL (Grill et al., 2020), which do not make use of any sort of class assignment (pseudo or otherwise).

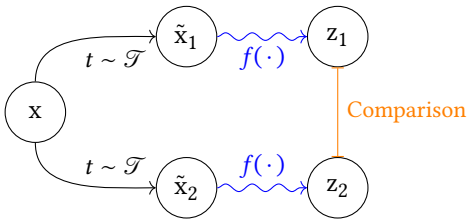

Fig. 4. A chart showing how general pairwise consistency regularization works. Generally, samples are drawn from $x \sim \mathcal{D}$, where a series of transformations $t \sim \mathcal{T}$ transforms x into a pair of unequally augmented samples $\tilde{x}_1, \tilde{x}_2$. This pair of samples is then projected into a latent representation by some projection function $f(\cdot)$ resulting in latent vectors z_1, z_2. Typically during consistency regularization, the loss function will be structured to minimize the distance from z_1, z_2 from the same sample x and maximize the distance from other sample representations within a minibatch.

In a stricter sense, there are approaches such as *Scaling Open-Vocabulary Object Detection* (Minderer et al., 2023) which make use of actual predicted class labels that are relevant to a future task. This is not a classification task but rather an object detection task, however the CLIP model is used to give weak supervision (pseudo-labels) for bounding boxes and this weak supervision is used as a pre-training step. This kind of weak supervision is common in open-vocabulary settings such as that used to train CLIP, however in the case of CLIP training that weak supervision came from humans not any model outputs.

Limitations. In order to perform consistency regularization the training algorithm has to define between which instances model predictions should be consistent. Typically, consistency is enforced between augmented versions of the same image or between different representations of the same data point. Effective consistency regularization relies on being able to define transformations to which the content of an image is invariant. This may pose a limitation to specialized data domains for which standard data augmentation techniques to not apply. Another possible failure for consistency regularization is representation collapse. It is possible that a model learns to predict the same output for every input which leads to very good consistency but very poor representation quality.

4.2 Knowledge Distillation

Knowledge distillation (Hinton et al., 2015) describes the process of transferring the knowledge of a larger network into a smaller one. The significant attention brought to this technique has yielded a variety of techniques (Gou et al., 2021). In this assessment we are concerned with *response-based* knowledge distillation; i.e., techniques for knowledge distillation that transfer knowledge through pseudo-labels assigned by the larger model. For a more comprehensive treatment of knowledge distillation see (Gou et al., 2021).

The formulation in Hinton et al. (2015) is a response-based framework which is semi-supervised. The original larger model is trained in a supervised way on a classification dataset. The process of distilling knowledge into the smaller model is semi-supervised in general, although if only the model and instances are present without labels the distillation can still occur with only self-supervision. The model is trained to minimize a loss function that is composed of a supervised term measuring the disagreement with the true labels of the examples, and another loss term which measures the disagreement with the prediction made by the larger model. This second term is clearly making use of PL.

As mentioned in Caron et al. (2021), the DINO framework for self-supervised learning is interpreted as a framework for knowledge distillation between a student model and a mean teacher (a teacher model which is an average of the student model's weights). This is self-distillation with *no labels* which is entirely based on PL. DINO can be seen as a special case of the *replace one branch* (Duval et al., 2023) approach to knowledge distillation with a slight modification of including a mean teacher. In fact, this is exactly how the smaller published DINO models are created. Rather than training the models from scratch, the smaller architectures' parameters are trained by holding the parameters of the larger model constant and distilling their knowledge from them into the smaller model using the same DINO framework which was used to train the larger model. This approach is broadly applicable to the above self-supervised frameworks and others which may not utilize PL but have a similar two-branch training structure.

In all cases a fuzzy partition parameterized by a neural network is inferred from data and available at the beginning of the distillation process. This is the teacher model, or the model which is being distilled. In the case of Hinton's original formulation this fuzzy partition clearly corresponds to division of the input space into classes informed by real labels. It is also possible to incorporate real labels in the distillation process in that formulation, however pseudo-labels are still used in the training process. in fact, the labels generated by the model and given to the student model are identical to those utilized in traditional pseudo-labeling approaches such as Lee (2013). In the case of DINO and RoB the labels do not necessarily correspond to any ground truth quantity, but the loss function is identical. DINO distillation and RoB also define fuzzy partitions of the input space into categories and the goal of learning in the knowledge distillation regime is the same: minimize the average cross entropy over a set of unlabeled instances and their associated pseudo-labels between the student and the teacher.

Limitations. Usually to perform knowledge distillation one needs access to the distribution on which the model to be distilled was trained. This may not be realistic if that model was trained on a proprietary dataset or one so large that distilling over it is prohibitively expensive.

5 Commonalities Between Discussed Methods

We have covered a variety of ways in which pseudo-labeling is applied in computer vision algorithms. We have connected these ideas with our interpretation of pseudo-labels. In this section we identify commonalities between the methods that we have discussed so far. This will then illuminate the directions for future work, which we present in the next section.

Filtering the Data. One of the main arguments presented in CLIP (Radford et al., 2021) is the extensive dataset filtering methodology, and even subsequent works such as ALIGN (Jia et al., 2021) still perform some dataset filtering. Such rigorous dataset construction seems to be the norm now with the LAION dataset using the CLIP pre-trained model to filter out noisy image-caption pairs (Schuhmann et al., 2022), and Xu et al. (2024) argue the dataset curation was an essential aspect of CLIP's success. Proprietary pre-filtered datasets such as JFT and IG-1B and open-source datasets such as CounterAnimal serve as an integral component in training state-of-the-art models' self-supervised and semi-supervised learning (Pham et al., 2021; Radford et al., 2021; Caron et al., 2021; Xie et al., 2020b; Wang et al., 2024). In knowledge distillation, maintaining the full model's performance requires some

access to the training data; without it, we must resort to techniques like data-free knowledge distillation (Lopes et al., 2017) or reconstructing a similar dataset from open data such as LAION-5B (Schuhmann et al., 2022). Thus, strategic dataset filtering emerges as a key enabler for effective pseudo-labeling.

Establishing a Curriculum. Curriculum learning (Bengio et al., 2009) is an approach that was originally proposed for supervised learning, but it is often used as the justification for metric-based selection procedures for PL (as discussed in Section 3.1.3). Outside SSL, curriculum approaches have received a small amount of attention recently in knowledge distillation literature with curricula inspired by the training trajectory of the heavier model (Jin et al., 2019), and by the temperature used during training (Li et al., 2023). Curriculum learning has also been applied in self-supervised learning to learn representations that are robust to spurious correlations (Zhu et al., 2023). The curricula proposed for knowledge distillation and self-supervised learning are practically interchangeable when using an appropriate framework such as RoB (Duval et al., 2023) and DINO (Caron et al., 2021) respectively. Though the connection is perhaps looser, the choice of a curriculum is very important for good self-training and co-training performance. There is similarly no practical reason the curricula proposed for self-training could not be applied to knowledge distillation or self-training and vice-versa.

Data Augmentation Strategies. Data augmentations typically perturb the input instances without altering their ground truth assignments, and strategies for this are varied and domain-dependent. All forms of data augmentation can be seen as providing pseudo-labels, as the original sample label is assigned to a new (augmented) sample. Rather than communicating the prediction of some machine learning model, these pseudo-labels instill an inductive bias. Outside of limited cases such as horizontal flips of most images (which can be known to not change the underlying class), augmentations such as MixUp (Zhang et al., 2018), the *-Match family (Berthelot et al., 2019b; Sohn et al., 2020; Berthelot et al., 2019a; Zhang et al., 2021), or even RandAugment (Cubuk et al., 2019) can meaningfully change or obscure class information. This is critical as they define the invariance the model is to learn and makes the most of limited supervised data.

Soft vs. Hard Pseudo-Labels. The dominant strategy is in line with the spirit of EM, where choosing hard labels extends the margin of the decision boundary. Some works report better performance when using soft labels Xie et al. (2020a), but other works take precautions to avoid trivial solutions when using soft labels Yi et al. (2022); not to mention the challenges of using soft labels (and thus weaker gradients) for larger model architectures. Conversely, hard labels can be sampled from a discrete distribution described by the output vector of a model yielding an assignment less aligned with the ground truth, but which strikes a desirable learning process exploration / exploitation trade-off (e.g., Pham et al. (2021)). Interestingly, this hard-label choice is not nearly as popular in the self-supervised or knowledge distillation areas where the argument usually follows from an argument centered around smoothing a distribution rather than performing EM. Despite this, such techniques typically make use of a large temperature parameter to "harden" the label output by the teacher (Chen et al., 2020; Caron et al., 2021; Li et al., 2022; Hinton et al., 2015; Wen et al., 2021; Li et al., 2023).

Updating the Teacher. There are a variety of strategies that are employed to update the teacher model from which knowledge is being transferred. In self-supervised settings a teacher is usually a model updated as the EMA of the student's weights from the last several steps (Grill et al., 2020; Caron et al., 2021; Li et al., 2022). Conversely, in semi-supervised learning the teacher's weights are usually the product of a separate optimization—either through meta-learning (Pham et al., 2021) or a previous weakly-supervised training iteration (Xie et al., 2020b; McLachlan, 1975; Blum and Mitchell, 1998; Hady and Schwenker, 2008; Yu et al., 2021; Chen et al., 2022; Zhou et al., 2020). Acknowledging the weakness of the teacher's labels within the knowledge distillation framework, one can even imagine altering teacher predictions to produce a better student (Wen et al., 2021). In all pseudo-labeling cases it is useful to consider how (if at all) the teacher should be updated.

Self-Supervised Regularization. In practice, there are far more unlabeled samples than labeled samples: for example, open source NLP/CV datasets contain billions of un- or weakly-curated samples (Schuhmann et al., 2022; Computer, 2023). Typically, representations are learned first on unsupervised data and then are transferred to a downstream task. In the case of full-parameter finetuning, however, it is possible for a network to 'forget' valuable information learned during pre-training while it is being finetuned. An important type of regularization may be to include a semi-supervised objective in the loss such as NT-Xent (Chen et al., 2020) or the BYOL (Grill et al., 2020) loss function; preventing a representation space from collapsing or to allow learning a representation space jointly with the classification objective.

6 Open Directions

Having identified the ways in which these different algorithms which use pseudo-labels are similar, we are now prepared to discuss unexplored directions. Below we outline what we believe to be some of the most promising open directions when one tries to exploit these commonalities between the different forms of pseudo-labeling.

Dataset Filtering. There is significant focus on dataset filtering in the self-supervised learning literature, however this remains relatively unexplored in semi-supervised learning and knowledge distillation. It seems intuitive that dataset subsetting has the potential to expedite knowledge distillation, particularly when training data for the model to be distilled is unavailable. Manipulating the unlabeled data distribution so that it is closer to the training distribution has the potential to increase the value of that unlabeled data. In semi-supervised learning, the typical method of evaluating the performance of algorithms is to hold out a subset of the labels from a balanced dataset. This yields an idealized unlabeled distribution which is not available in most practical scenarios. Evaluations performed this way ignore the cases of class imbalance and unknown class distributions which are inevitable for uncurated unlabeled sets. Semi-supervised learning algorithms are vulnerable to differences in distribution between the unlabeled and labeled sets, and they are vulnerable. Automated data filtering methods similar to Vo et al. (2024) are an exciting future direction for semi-supervised learning as they have the potential to improve the unlabeled data distribution without human intervention.

Establishing a Useful Curriculum. Establishing an effective curriculum is an open direction for all of these pseudo-labeling methods. In Section 3.1.3 we identified methods of semi-supervised learning which leverage training curricula to learn to label "easy" instances before "difficult" instances. These ideas were motivated by entropy minimization and rely on selecting pseudo-labels based on confidence. Model confidence is a common selection criterion for pseudo-labels, but as we established in Section 3.1.3 model calibration is not calibrated in general. We see the application of uncertainty quantification or calibration methods to the construction of curricula for semi-supervised learning as a promising future direction. For unsupervised or self-supervised pseudo-labeling methods which do no have a readily available notion of confidence the creation of a curriculum based on instance loss is also unexplored. Curriculum training broadly offers the potential to capture patterns in underrepresented regions of the training data which is valuable for any models trained on uncurated data.

Direct Initialization of Models. While models can be initialized by randomly sampling the weights from an initialization distribution, they can also be initialized using an intelligently chosen set of weights. A large number of pre-trained "foundation" models for vision have been produced which can be compressed using low-rank compression Yu et al. (2017); Xiao et al. (2023), neuron clustering Cho et al. (2022), or neuron selection Samragh et al. (2023). Knowledge distillation is sometimes leveraged to compress foundation models Caron et al. (2021); Duval et al. (2023). Direct initialization has the potential to "jump-start" this process. In semi-supervised learning student-teacher algorithms could also benefit from using compressed weights of the teacher to directly initialize the student. These methods of directly initializing neural network weights to reduce the cost of expensive pseudo-labeling training methods have so far been unexplored.

Self-Supervised Regularization of Semi-Supervised Learning. In semi-supervised learning and knowledge distillation there is a risk of representation collapse of the learner. One recent method, *self-supervised semi-supervised learning* (Berthelot et al., 2019a), incorporates a self-supervised loss term to prevent that representation collapse. Despite the simplicity they are able to improve on supervised baselines. Their approach and others such as Rothenberger and Diochnos (2023) show that the benefits for learning with few labels from pseudo-labeling-based self-supervised learning and pseudo-labeling-based semi-supervised learning are complementary. Combining the benefits of knowledge distillation and semi-supervised learning with the strong performance of representations learned with self-supervised loss terms is an exciting avenue of research for pseudo-labeling.

Meta Learning to Determine Pseudo-Label Assignment. In Section 3.4 we identified several methods for semi-supervised learning which formulate pseudo-label assignment as a meta-learning problem or bi-level optimization. Several recent pseudo-labeling methods for semi-supervised learning Pham et al. (2021); Rothenberger and Diochnos (2023); Heidari and Guo (2025); Heidari et al. (2024) leverage bi-level frameworks for pseudo-label assignment. RLGSSL Heidari et al. (2024) we find to be particular interesting because it interprets this bi-level optimization as reinforcement learning which opens many new potential directions for exploration based on existing work in reinforcement learning. Developing new ways of formulating and optimizing these bi-level frameworks is an exciting direction for self-supervised learning. To our knowledge bi-level optimization has not been applied in this way to either knowledge distillation or self-supervised learning. Due to the promising results of pseudo-labeling methods based on bi-level optimizations for semi-supervised learning we believe similar algorithms can find success in the areas of knowledge distillation and self-supervised learning.

7 Conclusion

In Section 1 we gave an overview of theoretical motivations for pseudo-labeling. We provided a taxonomy of methods for pseudo-labeling in these areas and discussed their relation to the important motivating ideas of sample scheduling, label propagation, weak supervision, consistency regularization, multi-model learning, and knowledge distillation. This taxonomy was shown in Figure 1. We provided a comparison of methods applied to semi-supervised image classification; these comparative results were shown in Table 1. In Section 2 we introduced a definition for pseudo-labels. We showed how this definition allows for a unified interpretation of methods from semi-supervised learning and self-supervised learning. In Section 3 we presented many different methods for semi-supervised learning that make use of pseudo-labels. In Section 4 we showed how pseudo-labels are used in unsupervised and self-supervised ways. We discussed commonalities that we observed between different types of pseudo-labeling algorithms in Section 5. In Section 6 we discussed open directions that were illuminated by our definition of pseudo-labels. Finally, in Table 2 in Appendix A we provide a glossary of the different techniques that are shown in the taxonomy of Figure 1. Table 2 provides the paper that proposed each technique and the section in which they appear in this review.

We find that there are many exciting new opportunities for research. The prospect of applying dataset subsetting methods on data collected from the wild is particularly exciting, and so is the application of reinforcement learning algorithms/bi-level optimization frameworks and uncertainty quantification. We are excited to see how new applications of pseudo-labels continue to shape the computer vision landscape.

Acknowledgments

This material is based upon work supported by the U.S. National Science Foundation under Grant No. RISE-2019758. This work is part of the NSF AI Institute for Research on Trustworthy AI in Weather, Climate, and Coastal Oceanography (NSF AI2ES). The authors would like to thank Dr. Kerri Cahoy for providing valuable input on an earlier version of this work.

References

Massih-Reza Amini, Vasilii Feofanov, Loïc Pauletto, Lies Hadjadj, Emilie Devijver, and Yury Maximov. 2025. Self-training: A survey. *Neurocomputing* 616 (2025), 128904. https://doi.org/10.1016/J.NEUCOM.2024.128904

Dana Angluin and Philip D. Laird. 1987. Learning From Noisy Examples. *Mach. Learn.* 2, 4 (1987), 343–370. https://doi.org/10.1007/BF00116829

Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. 2020. Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020*. IEEE, , 1–8. https://doi.org/10.1109/IJCNN48605.2020.9207304

Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Grégoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. 2023. A Cookbook of Self-Supervised Learning. *CoRR* abs/2304.12210 (2023), 71. https://doi.org/10.48550/ARXIV.2304.12210 arXiv:2304.12210

Marcos Barcina-Blanco, Jesus L. Lobo, Pablo Garcia-Bringas, and Javier Del Ser. 2024. Managing the unknown: a survey on Open Set Recognition and tangential areas. arXiv:2312.08785 [cs.LG] https://arxiv.org/abs/2312.08785

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009 (ACM International Conference Proceeding Series, Vol. 382)*, Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman (Eds.). ACM, , 41–48. https://doi.org/10.1145/1553374.1553380

David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. 2019a. ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring. *CoRR* abs/1911.09785 (2019), 13. arXiv:1911.09785 http://arxiv.org/abs/1911.09785

David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. 2019b. MixMatch: A Holistic Approach to Semi-Supervised Learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). , , 5050–5060. https://proceedings.neurips.cc/paper/2019/hash/1cd138d0499a68f4bb72bee04bbec2d7-Abstract.html

Avrim Blum and Tom M. Mitchell. 1998. Combining Labeled and Unlabeled Data with Co-Training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998, Madison, Wisconsin, USA, July 24-26, 1998*, Peter L. Bartlett and Yishay Mansour (Eds.). ACM, , 92–100. https://doi.org/10.1145/279943.279962

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight Uncertainty in Neural Network. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015 (JMLR Workshop and Conference Proceedings, Vol. 37)*, Francis R. Bach and David M. Blei (Eds.). JMLR.org, , 1613–1622. http://proceedings.mlr.press/v37/blundell15.html

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep Clustering for Unsupervised Learning of Visual Features. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIV (Lecture Notes in Computer Science, Vol. 11218)*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer, , 139–156. https://doi.org/10.1007/978-3-030-01264-9_9

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin

(Eds.). , , 13. https://proceedings.neurips.cc/paper/2020/hash/70feb62b69f16e0238f741fab228fec2-Abstract.html

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging Properties in Self-Supervised Vision Transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, , 9630–9640. https://doi.org/10.1109/ICCV48922.2021.00951

Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. 2020. Curriculum Labeling: Self-paced Pseudo-Labeling for Semi-Supervised Learning. *CoRR* abs/2001.06001 (2020), 13. arXiv:2001.06001 https://arxiv.org/abs/2001.06001

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien (Eds.). 2006. *Semi-Supervised Learning*. The MIT Press, . https://doi.org/10.7551/MITPRESS/9780262033589.001.0001

Mingcai Chen, Yuntao Du, Yi Zhang, Shuwei Qian, and Chongjun Wang. 2022. Semi-supervised Learning with Multi-Head Co-Training. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, , 6278–6286. https://doi.org/10.1609/AAAI.V36I6.20577

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, , 1597–1607. http://proceedings.mlr.press/v119/chen20j.html

Minsik Cho, Keivan Alizadeh-Vahid, Saurabh Adya, and Mohammad Rastegari. 2022. DKM: Differentiable k-Means Clustering Layer for Neural Network Compression. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, , 19. https://openreview.net/forum?id=J_F_qqCE3Z5

Together Computer. 2023. RedPajama: an Open Dataset for Training Large Language Models. https://github.com/togethercomputer/RedPajama-Data.

Ekin Dogus Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. 2018. AutoAugment: Learning Augmentation Policies from Data. *CoRR* abs/1805.09501 (2018), 14. arXiv:1805.09501 http://arxiv.org/abs/1805.09501

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. 2019. RandAugment: Practical data augmentation with no separate search. *CoRR* abs/1909.13719 (2019), 13. arXiv:1909.13719 http://arxiv.org/abs/1909.13719

Quentin Duval, Ishan Misra, and Nicolas Ballas. 2023. A Simple Recipe for Competitive Low-compute Self supervised Vision Models. *CoRR* abs/2301.09451 (2023), 15. https://doi.org/10.48550/ARXIV.2301.09451 arXiv:2301.09451

Benoît Frénay and Michel Verleysen. 2014. Classification in the Presence of Label Noise: A Survey. *IEEE Trans. Neural Networks Learn. Syst.* 25, 5 (2014), 845–869. https://doi.org/10.1109/TNNLS.2013.2292894

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016 (JMLR Workshop and Conference Proceedings, Vol. 48)*, Maria-Florina Balcan and Kilian Q. Weinberger (Eds.). JMLR.org, , 1050–1059. http://proceedings.mlr.press/v48/gal16.html

Bin-Bin Gao, Chao Xing, Chen-Wei Xie, Jianxin Wu, and Xin Geng. 2017. Deep Label Distribution Learning with Label Ambiguity. *IEEE Transactions on Image Processing* 26, 6 (June 2017), 2825–2838. https://doi.org/10.1109/TIP.2017.2689998 arXiv:1611.01731 [cs]

Aurélien Géron. 2022. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow.* " O'Reilly Media, Inc.", .

Jacob Goldberger and Ehud Ben-Reuven. 2017. Training deep neural-networks using a noise adaptation layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference*

Track Proceedings. OpenReview.net, , 9. https://openreview.net/forum?id=H12GRgcxg

Sally A. Goldman and Yan Zhou. 2000. Enhancing Supervised Learning with Unlabeled Data. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, Pat Langley (Ed.). Morgan Kaufmann, , 327–334.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge Distillation: A Survey. *Int. J. Comput. Vis.* 129, 6 (2021), 1789–1819. https://doi.org/10.1007/S11263-021-01453-Z

Alex Graves. 2011. Practical Variational Inference for Neural Networks. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger (Eds.). , , 2348–2356. https://proceedings.neurips.cc/paper/2011/hash/7eb3c8be3d411e8ebfab08eba5f49632-Abstract.html

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. 2020. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). , , 14. https://proceedings.neurips.cc/paper/2020/hash/f3ada80d5c4ee70142b17b8192b2958e-Abstract.html

Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R. Scott, and Dinglong Huang. 2018. CurriculumNet: Weakly Supervised Learning from Large-Scale Web Images. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X (Lecture Notes in Computer Science, Vol. 11214)*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer, , 139–154. https://doi.org/10.1007/978-3-030-01249-6_9

Christian Haase-Schütz, Rainer Stal, Heinz Hertlein, and Bernhard Sick. 2020. Iterative Label Improvement: Robust Training by Confidence Based Filtering and Dataset Partitioning. In *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*. IEEE, , 9483–9490. https://doi.org/10.1109/ICPR48806.2021.9411918

Mohamed Farouk Abdel Hady and Friedhelm Schwenker. 2008. Co-training by Committee: A New Semi-supervised Learning Framework. In *Workshops Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*. IEEE Computer Society, , 563–572. https://doi.org/10.1109/ICDMW.2008.27

Bo Han, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W. Tsang, James T. Kwok, and Masashi Sugiyama. 2020. A Survey of Label-noise Representation Learning: Past, Present and Future. *CoRR* abs/2011.04406 (2020), 24. arXiv:2011.04406 https://arxiv.org/abs/2011.04406

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). , , 8536–8546. https://proceedings.neurips.cc/paper/2018/hash/a19744e268754fb0148b017647355b7b-Abstract.html

Marzi Heidari and Yuhong Guo. 2025. Bi-Level Optimization for Pseudo-Labeling Based Semi-Supervised Learning. https://openreview.net/forum?id=AEi2wyAMyb

Marzi Heidari, Hanping Zhang, and Yuhong Guo. 2024. Reinforcement Learning-Guided Semi-Supervised Learning. arXiv:2405.01760 [cs.LG] https://arxiv.org/abs/2405.01760

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. *CoRR* abs/1503.02531 (2015), 9. arXiv:1503.02531 http://arxiv.org/abs/1503.02531

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. 2019. Label Propagation for Deep Semi-Supervised Learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, , 5070–5079. https://doi.org/10.1109/CVPR.2019.00521

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor. 2023. *An introduction to statistical learning: With applications in python.* Springer Nature, .

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, , 4904–4916. http://proceedings.mlr.press/v139/jia21b.html

Xiao Jin, Baoyun Peng, Yichao Wu, Yu Liu, Jiaheng Liu, Ding Liang, Junjie Yan, and Xiaolin Hu. 2019. Knowledge Distillation via Route Constrained Optimization. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, , 1345–1354. https://doi.org/10.1109/ICCV.2019.00143

Longlong Jing and Yingli Tian. 2021. Self-Supervised Visual Feature Learning With Deep Neural Networks: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 11 (2021), 4037–4058. https://doi.org/10.1109/TPAMI.2020.2992393

Patrick Kage and Pavlos Andreadis. 2021. Class Introspection: A Novel Technique for Detecting Unlabeled Subclasses by Leveraging Classifier Explainability Methods. *CoRR* abs/2107.01657 (2021), 10. arXiv:2107.01657 https://arxiv.org/abs/2107.01657

Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. 2019. Revisiting Self-Supervised Visual Representation Learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, , 1920–1929. https://doi.org/10.1109/CVPR.2019.00202

Samuli Laine and Timo Aila. 2017. Temporal Ensembling for Semi-Supervised Learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, , 13. https://openreview.net/forum?id=BJ6oOfqge

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). , , 6402–6413. https://proceedings.neurips.cc/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html

Dong-Hyun Lee. 2013. Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. *ICML 2013 Workshop: Challenges in Representation Learning (WREPL)* 3, 2 (July 2013), 896.

Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. 2022. Efficient Self-supervised Vision Transformers for Representation Learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, , 27. https://openreview.net/forum?id=fVu3o-YUGQK

Zheng Li, Xiang Li, Lingfeng Yang, Borui Zhao, Renjie Song, Lei Luo, Jun Li, and Jian Yang. 2023. Curriculum Temperature for Knowledge Distillation. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, Brian Williams, Yiling Chen, and Jennifer Neville (Eds.). AAAI Press, , 1504–1512. https://doi.org/10.1609/AAAI.V37I2.25236

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. 2021. Energy-Based Out-of-distribution Detection. https://doi.org/10.48550/arXiv.2010.03759 arXiv:2010.03759 [cs]

Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. 2017. Data-Free Knowledge Distillation for Deep Neural Networks. *CoRR* abs/1710.07535 (2017), 8. arXiv:1710.07535 http://arxiv.org/abs/1710.07535

Christos Louizos and Max Welling. 2016. Structured and Efficient Variational Deep Learning with Matrix Gaussian Posteriors. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016 (JMLR Workshop and Conference Proceedings, Vol. 48)*, Maria-Florina Balcan and Kilian Q. Weinberger (Eds.). JMLR.org, , 1708–1716. http://proceedings.mlr.press/v48/louizos16.html

G. J. McLachlan. 1975. Iterative Reclassification Procedure for Constructing an Asymptotically Optimal Rule of Allocation in Discriminant Analysis. *J. Amer. Statist. Assoc.* 70, 350 (1975), 365–369. http://www.jstor.org/stable/2285824

Matthias Minderer, Alexey A. Gritsenko, and Neil Houlsby. 2023. Scaling Open-Vocabulary Object Detection. *CoRR* abs/2306.09683 (2023), 22. https://doi.org/10.48550/ARXIV.2306.09683 arXiv:2306.09683

Tom M. Mitchell. 1997. *Machine learning, International Edition.* McGraw-Hill, . https://www.worldcat.org/oclc/61321007

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2019. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 8 (2019), 1979–1993. https://doi.org/10.1109/TPAMI.2018.2858821

Kamal Nigam, Andrew Kachites Mccallum, Sebastian Thrun, and Tom Mitchell. 2000. Text Classification from Labeled and Unlabeled Documents Using EM. *Machine Learning* 39, 2 (May 2000), 103–134. https://doi.org/10.1023/A:1007692713085

Yassine Ouali, Céline Hudelot, and Myriam Tami. 2020. An Overview of Deep Semi-Supervised Learning. *CoRR* abs/2006.05278 (2020), 43. arXiv:2006.05278 https://arxiv.org/abs/2006.05278

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making Deep Neural Networks Robust to Label Noise: A Loss Correction Approach. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, , 2233–2241. https://doi.org/10.1109/CVPR.2017.240

Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V. Le. 2021. Meta Pseudo Labels. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, , 11557–11568. https://doi.org/10.1109/CVPR46437.2021.01139

Nitin Namdeo Pise and Parag A. Kulkarni. 2008. A Survey of Semi-Supervised Learning Methods. In *2008 International Conference on Computational Intelligence and Security, CIS 2008, 13-17 December 2008, Suzhou, China, Volume 2, Workshop Papers*. IEEE Computer Society, , 30–34. https://doi.org/10.1109/CIS.2008.204

V. Jothi Prakash and L. M. Nithya. 2014. A Survey on Semi-Supervised Learning Techniques. *CoRR* abs/1402.4645 (2014), 5. arXiv:1402.4645 http://arxiv.org/abs/1402.4645

Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan L. Yuille. 2018. Deep Co-Training for Semi-Supervised Image Recognition. *CoRR* abs/1803.05984 (2018), 17. arXiv:1803.05984 http://arxiv.org/abs/1803.05984

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, , 8748–8763. http://proceedings.mlr.press/v139/radford21a.html

Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S. Rawat, and Mubarak Shah. 2021. In Defense of Pseudo-Labeling: An Uncertainty-Aware Pseudo-label Selection Framework for Semi-Supervised Learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, , 20. https://openreview.net/forum?id=-ODN6SbiUU

Jay C. Rothenberger and Dimitrios I. Diochnos. 2023. Meta Co-Training: Two Views are Better than One. *CoRR* abs/2311.18083 (2023), 16. https://doi.org/10.48550/ARXIV.2311.18083 arXiv:2311.18083

Enrique H. Ruspini. 1969. A new approach to clustering. *Information and Control* 15, 1 (1969), 22–32. https://doi.org/10.1016/S0019-9958(69)90591-9

Mohammad Samragh, Mehrdad Farajtabar, Sachin Mehta, Raviteja Vemulapalli, Fartash Faghri, Devang Naik, Oncel Tuzel, and Mohammad Rastegari. 2023. Weight subcloning: direct initialization of transformers using larger pretrained ones. *CoRR* abs/2312.09299 (2023), 10. https://doi.org/10.48550/ARXIV.2312.09299 arXiv:2312.09299

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. 2022. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.). , , 17. http://papers.nips.cc/paper_files/paper/2022/hash/a1859debfb3b59d094f3504d5ebb6c25-Abstract-Datasets_and_Benchmarks.html

Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, . http://www.cambridge.org/de/academic/subjects/computer-science/pattern-recognition-and-machine-learning/understanding-machine-learning-theory-algorithms

Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng Ma, Xiaoyu Tao, and Nanning Zheng. 2018. Transductive Semi-Supervised Deep Learning Using Min-Max Features. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part V (Lecture Notes in Computer Science, Vol. 11209)*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer, , 311–327. https://doi.org/10.1007/978-3-030-01228-1_19

Connor Shorten and Taghi M. Khoshgoftaar. 2019. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* 6 (2019), 60. https://doi.org/10.1186/S40537-019-0197-0

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). , , 13. https://proceedings.neurips.cc/paper/2020/hash/06964dce9addb1c5cb5d6e3d9838f733-Abstract.html

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2023. Learning From Noisy Labels With Deep Neural Networks: A Survey. *IEEE Trans. Neural Networks Learn. Syst.* 34, 11 (2023), 8135–8153. https://doi.org/10.1109/TNNLS.2022.3152527

Shiliang Sun, Feng Jin, and Wenting Tu. 2011. View Construction for Multi-view Semi-supervised Learning. In *Advances in Neural Networks - ISNN 2011 - 8th International Symposium on Neural Networks, ISNN 2011, Guilin, China, May 29-June 1, 2011, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 6675)*, Derong Liu, Huaguang Zhang, Marios M. Polycarpou, Cesare Alippi, and Haibo He (Eds.). Springer, , 595–601. https://doi.org/10.1007/978-3-642-21105-8_69

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). , , 1195–1204. https://proceedings.neurips.cc/paper/2017/hash/68053af2923e00204c3ca7c6a3150cf7-Abstract.html

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. 2015. Learning from Massive Noisy Labeled Data for Image Classification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Boston, MA, USA, 2691–2699. https://doi.org/10.1109/CVPR.2015.7298885

Jesper E. van Engelen and Holger H. Hoos. 2020. A survey on semi-supervised learning. *Mach. Learn.* 109, 2 (2020), 373–440. https://doi.org/10.1007/S10994-019-05855-6

Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Arno Solin, Yoshua Bengio, and David Lopez-Paz. 2022. Interpolation consistency training for semi-supervised learning. *Neural Networks* 145 (Jan. 2022), 90–106. https://doi.org/10.1016/j.neunet.2021.10.008

Huy V. Vo, Vasil Khalidov, Timothée Darcet, Théo Moutakanni, Nikita Smetanin, Marc Szafraniec, Hugo Touvron, Camille Couprie, Maxime Oquab, Armand Joulin, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. 2024. Automatic Data Curation for Self-Supervised Learning: A Clustering-Based Approach. *arXiv:2405.15613* (2024).

Erik Wallin, Lennart Svensson, Fredrik Kahl, and Lars Hammarstrand. 2023. Improving Open-Set Semi-Supervised Learning with Self-Supervision. arXiv:2301.10127 [cs.LG] https://arxiv.org/abs/2301.10127

Erik Wallin, Lennart Svensson, Fredrik Kahl, and Lars Hammarstrand. 2024. ProSub: Probabilistic Open-Set Semi-Supervised Learning with Subspace-Based Out-of-Distribution Detection. arXiv:2407.11735 [cs.LG] https://arxiv.org/abs/2407.11735

Jiao Wang, Siwei Luo, and Xianhua Zeng. 2008. A random subspace method for co-training. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2008, part of the IEEE World Congress on Computational Intelligence, WCCI 2008, Hong Kong, China, June 1-6, 2008*. IEEE, , 195–200. https://doi.org/10.1109/IJCNN.2008.4633789

Qizhou Wang, Bo Han, Tongliang Liu, Gang Niu, Jian Yang, and Chen Gong. 2021. Tackling Instance-Dependent Label Noise via a Universal Probabilistic Model. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, , 10183–10191. https://doi.org/10.1609/AAAI.V35I11.17221

Qizhou Wang, Bo Han, Tongliang Liu, Gang Niu, Jian Yang, and Chen Gong. 2022. Tackling Instance-Dependent Label Noise via a Universal Probabilistic Model. https://doi.org/10.48550/arXiv.2101.05467 arXiv:2101.05467 [cs]

Qizhou Wang, Bo Han, Yang Liu, Chen Gong, Tongliang Liu, and Jiming Liu. 2025. W-DOE: Wasserstein Distribution-Agnostic Outlier Exposure . *IEEE Transactions on Pattern Analysis & Machine Intelligence* 47, 05 (May 2025), 3530–3545. https://doi.org/10.1109/TPAMI.2025.3531000

Qizhou Wang, Yong Lin, Yongqiang Chen, Ludwig Schmidt, Bo Han, and Tong Zhang. 2024. Do CLIP Models Always Generalize Better than ImageNet Models?. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=wWyumwEYV8

Wei Wang and Zhi-Hua Zhou. 2013. Co-Training with Insufficient Views. In *Asian Conference on Machine Learning, ACML 2013, Canberra, ACT, Australia, November 13-15, 2013 (JMLR Workshop and Conference Proceedings, Vol. 29)*, Cheng Soon Ong and Tu Bao Ho (Eds.). JMLR.org, , 467–482. http://proceedings.mlr.press/v29/Wang13b.html

Yulin Wang, Jiayi Guo, Shiji Song, and Gao Huang. 2020. Meta-Semi: A Meta-learning Approach for Semi-supervised Learning. *CoRR* abs/2007.02394 (2020), 18. arXiv:2007.02394 https://arxiv.org/abs/2007.02394

Tiancheng Wen, Shenqi Lai, and Xueming Qian. 2021. Preparing lessons: Improve knowledge distillation with better supervision. *Neurocomputing* 454 (2021), 25–33. https://doi.org/10.1016/J.NEUCOM.2021.04.102

Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. 2019. Are Anchor Points Really Indispensable in Label-Noise Learning?. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). , , 6835–6846. https://proceedings.neurips.cc/paper/2019/hash/9308b0d6e5898366a4a986bc33f3d3e7-Abstract.html

Jinqi Xiao, Chengming Zhang, Yu Gong, Miao Yin, Yang Sui, Lizhi Xiang, Dingwen Tao, and Bo Yuan. 2023. HALOC: Hardware-Aware Automatic Low-Rank Compression for Compact Neural Networks. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of*

Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023, Brian Williams, Yiling Chen, and Jennifer Neville (Eds.). AAAI Press, , 10464–10472. https://doi.org/10.1609/AAAI.V37I9.26244

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. 2020a. Unsupervised Data Augmentation for Consistency Training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). , , 20. https://proceedings.neurips.cc/paper/2020/hash/44feb0096faa8326192570788b38c1d1-Abstract.html

Qizhe Xie, Minh-Thang Luong, Eduard H. Hovy, and Quoc V. Le. 2020b. Self-Training With Noisy Student Improves ImageNet Classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, , 10684–10695. https://doi.org/10.1109/CVPR42600.2020.01070

Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. 2024. Demystifying CLIP Data. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, , 20. https://openreview.net/forum?id=5BCFlnfE1g

Yan Yan, Rómer Rosales, Glenn Fung, Subramanian Ramanathan, and Jennifer G. Dy. 2014. Learning from multiple annotators with varying expertise. *Mach. Learn.* 95, 3 (2014), 291–327. https://doi.org/10.1007/S10994-013-5412-1

Yan Yan, Zhongwen Xu, Ivor W. Tsang, Guodong Long, and Yi Yang. 2016. Robust Semi-Supervised Learning through Label Aggregation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, Dale Schuurmans and Michael P. Wellman (Eds.). AAAI Press, , 2244–2250. https://doi.org/10.1609/AAAI.V30I1.10276

Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. 2023. A Survey on Deep Semi-Supervised Learning. *IEEE Trans. Knowl. Data Eng.* 35, 9 (2023), 8934–8954. https://doi.org/10.1109/TKDE.2022.3220219

David Yarowsky. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *33rd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Cambridge, Massachusetts, USA, 189–196. https://doi.org/10.3115/981658.981684

Kun Yi, Guo-Hua Wang, and Jianxin Wu. 2022. PENCIL: Deep Learning with Noisy Labels. https://doi.org/10.48550/arXiv.2202.08436 arXiv:2202.08436 [cs]

Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, and Nguyen Quoc Viet Hung. 2021. Socially-Aware Self-Supervised Tri-Training for Recommendation. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, Feida Zhu, Beng Chin Ooi, and Chunyan Miao (Eds.). ACM, , 2084–2092. https://doi.org/10.1145/3447548.3467340

Xiyu Yu, Tongliang Liu, Xinchao Wang, and Dacheng Tao. 2017. On Compressing Deep Models by Low Rank and Sparse Decomposition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, , 67–76. https://doi.org/10.1109/CVPR.2017.15

Bodi Yuan, Jianyu Chen, Weidong Zhang, Hung-Shuo Tai, and Sara McMains. 2018. Iterative Cross Learning on Noisy Labels. In *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*. IEEE Computer Society, , 757–765. https://doi.org/10.1109/WACV.2018.00088

L.A. Zadeh. 1965. Fuzzy sets. *Information and Control* 8, 3 (1965), 338–353. https://doi.org/10.1016/S0019-9958(65)90241-X

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid Loss for Language Image Pre-Training. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, , 11941–11952. https://doi.org/10.1109/ICCV51070.2023.01100

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. 2021. FlexMatch: Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling. In *Advances in Neural*

Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). , , 18408–18419. https://proceedings.neurips.cc/paper/2021/hash/995693c15f439e3d189b06e89d145dd5-Abstract.html

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, , 13. https://openreview.net/forum?id=r1Ddp1-Rb

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. 2019. Be Your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, , 3712–3721. https://doi.org/10.1109/ICCV.2019.00381

Tianyi Zhou, Shengjie Wang, and Jeff A. Bilmes. 2020. Time-Consistent Self-Supervision for Semi-Supervised Learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, , 11523–11533. http://proceedings.mlr.press/v119/zhou20d.html

Yan Zhou and Sally A. Goldman. 2004. Democratic Co-Learning. In *16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004), 15-17 November 2004, Boca Raton, FL, USA*. IEEE Computer Society, , 594–602. https://doi.org/10.1109/ICTAI.2004.48

Zhi-Hua Zhou and Ming Li. 2005. Tri-Training: Exploiting Unlabeled Data Using Three Classifiers. *IEEE Trans. Knowl. Data Eng.* 17, 11 (2005), 1529–1541. https://doi.org/10.1109/TKDE.2005.186

Weicheng Zhu, Sheng Liu, Carlos Fernandez-Granda, and Narges Razavian. 2023. Making Self-supervised Learning Robust to Spurious Correlation via Learning-speed Aware Sampling. *CoRR* abs/2311.16361 (2023), 18. https://doi.org/10.48550/ARXIV.2311.16361 arXiv:2311.16361

Xiaojin Zhu and Zoubin Ghahramani. 2002. *Learning from Labeled and Unlabeled Data with Label Propagation*. Technical Report. Carnegie Mellon University, Pittsburgh, PA, USA.

Xiaojin Zhu and Andrew B. Goldberg. 2009. *Introduction to Semi-Supervised Learning*. Morgan & Claypool Publishers, . https://doi.org/10.2200/S00196ED1V01Y200906AIM006

A Reference Table for the Literature Shown in Figure 1

Table 2 has a mapping between the techniques that appear in Figure 1 (first column), with the appropriate references that are associated with the corresponding techniques (second column), and finally the section in this work where that technique is discussed.

Table 2. Figure 1 Label Map

Label	Reference	Section
BLOPL	Heidari and Guo (2025)	3.4
BYOL	Grill et al. (2020)	4.1
Class Introspection	Kage and Andreadis (2021)	3.2
Co-training	Blum and Mitchell (1998)	3.4
Cocommittee	Hady and Schwenker (2008)	3.4
DINO	Caron et al. (2021)	4.2
DL from NL	Tong Xiao et al. (2015)	3.2
DeepCluster	Caron et al. (2018)	4.1

Dem. Colearning	Zhou and Goldman (2004)	3.4
EsViT	Li et al. (2022)	4.1
FixMatch	Sohn et al. (2020)	3.1
FlexMatch	Zhang et al. (2021)	3.1
IDN	Wang et al. (2022)	3.2
ILI	Haase-Schütz et al. (2020)	3.2
Knowledge Distillation	Hinton et al. (2015)	4.2
LabelPropPL	Iscen et al. (2019)	3.2
MHCT	Chen et al. (2022)	3.4
MMF	Shi et al. (2018)	3.2
Meta Co-training	Rothenberger and Diochnos (2023)	3.4
MetaPL	Pham et al. (2021)	3.4
MixMatch	Berthelot et al. (2019b)	3.1
Noisy Student	Xie et al. (2020b)	3.4
OVOD	Minderer et al. (2023)	4.1
PENCIL	Yi et al. (2022)	3.2
PL (2013)	Lee (2013)	3
ProSub	Wallin et al. (2024)	3.2
RLGSSL	Heidari et al. (2024)	3.4
ROSSEL	Yan et al. (2016)	3.4
ReMixMatch	Berthelot et al. (2019a)	3.1
Replace One Branch	Duval et al. (2023)	4.2
SeFOSS	Wallin et al. (2023)	3.2
SwAV	Caron et al. (2020)	4.1
TCSSL	Zhou et al. (2020)	3.1
TriTraining	Zhou and Li (2005)	3.4
UDA	Xie et al. (2020a)	3.3
UPS	Rizve et al. (2021)	3.2
VAT	Miyato et al. (2019)	3.3
W-DOE	Wang et al. (2025)	3.2

Received 24 June 2024; accepted 15 November 2025